

BRUNET Magalie
LANÇON Guillaume
REYREAU Évelyne
SAUNAL Adrien



Mini – Projet

« Réalisation d'un prototype sur une architecture client/serveur pour la saisie de données géographiques à distance »



TABLE DES MATIÈRES

REMERCIEMENTS	3
INTRODUCTION.....	4
I. ETAT DES LIEUX.....	5
A. ANALYSE DE L'EXISTANT	5
1) Le matériel et les logiciels	5
2) Les données.....	5
B. MODELISATION DES DONNEES	6
1) Modèle Conceptuel de Données (MCD)	6
2) Modèle Logique de Données (MLD).....	6
II. DEFINITION DES BESOINS ET DES ATTENTES.....	7
A. NIVEAU TECHNIQUE :.....	7
1) Type de logiciel	7
2) Sécurisation des accès :	7
3) Administration.....	7
B. APPLICATION.....	7
1) Affichage des données de référence.	7
2) Saisie des données naturalistes.....	8
3) Consultation.....	8
4) Imports/exports	8
5) Données accessibles.....	8
C. DROIT DES DONNEES	9
1) Les données « Raster ».....	9
2) Les données « Vecteur ».....	10
III. DEROULEMENT DU PROJET	11
A. PLANNING	11
B. DOCUMENTS A REALISER.....	12
1) Guide d'installation et de maintenance du système sur le serveur	12
2) Guide explicatif de création de la base de données	12
3) Guide d'utilisation du logiciel client.....	12
IV. DEFINITION DE LA SOLUTION TECHNIQUE.....	13
A. ORGANISATION DU SYSTEME	13
1) Le système d'exploitation.....	13
2) Le SGBD	13
3) Le logiciel SIG client.....	14
4) L'organisation choisie répond-elle aux attentes du CREN ?	16
B. LES PERSPECTIVES DU PROJET	17
1) Évolution des choix techniques	17
2) Évolution du projet	17
V. BILAN	18
A. LES PROBLEMES RENCONTRES	18
B. DE NOUVEAUX ACQUIS TECHNIQUES ET PRATIQUES	18
SOURCES.....	19
ANNEXES	21
GUIDE D'INSTALLATION ET DE MAINTENANCE.....	22
GUIDE EXPLICATIF DE CRÉATION DE LA BASE DE DONNÉES	50
GUIDE D'UTILISATION DU LOGICIEL SIG	84

REMERCIEMENTS

Nous tenons à remercier tout d'abord nos deux Tuteurs : M **Laurent PONTCHARRAUD** du Conservatoire Régional des Espaces Naturels (CREN) de Midi-Pyrénées et M **Frédéric BARNAY** de Geo2i, pour nous avoir fourni un sujet intéressant et pour leurs aides.

Nous aimerions tout particulièrement remercier M **Laurent JÉGOU** ingénieur cartographe et enseignant à l'Université de Toulouse le Mirail au département de géographie, pour sa contribution, sa disponibilité, son aide et pour toutes les astuces qu'il nous a enseigné pour mener à bien notre mini-projet.

Nous tenons également à remercier l'ensemble des professeurs et des intervenants ayant participé à notre formation au sein du Master SIGMA.

Enfin, nous remercions l'ensemble des anonymes ayant participé à la réalisation de forums et de sites sur le Web. Ils ont ainsi contribué à la bonne mise en œuvre de notre projet par la consultation de leurs réponses.

INTRODUCTION

Les Conservatoires Régionaux des Espaces Naturels sont des associations Loi 1901 réunis au sein de « Espaces Naturels de France », la Fédération des Conservatoires d'Espaces Naturels. L'objet principal des ces organismes est la conservation de sites naturels d'intérêt écologique et biologique. En effet, le CREN a pour orientation de connaître, valoriser, protéger et gérer les espaces naturels remarquables.

Le CREN de Midi-Pyrénées, créé en 1988, recueille des données sur la faune et la flore de la région. Les données sont recueillies par quatre personnes localisées dans des villes différentes. Ainsi les administrateurs de données, Laurent PONTCHARRAUD et Pierre-Emmanuel RECH, se trouve à Toulouse tandis que Sylvain DEJEAN travaille à Foix (Ariège – 09) et Frédéric NERI à Brassac (Tarn – 81). Ces quatre personnes renseignent donc la même base de données, ce qui pose des problèmes pratiques de transmission et d'administration des données. En effet, il n'existe aucun système leur permettant d'y accéder à distance.

L'objectif de ce projet est de réfléchir à une solution permettant de résoudre les problèmes liés à la disparité des données en permettant aux utilisateurs concernés de travailler la base à distance.

A cet égard, le CREN a passé un contrat avec la société Geo2i (spécialisée dans les technologies de l'information géographique) afin de mettre en place un nouveau système de gestion et de consultation des données à distance basé sur de l'Open Source.

Les logiciels Open Source mettent à disposition leurs codes sources qui sont librement modifiables. Cette technologie garantit « *la qualité de la programmation, une maintenance facile, une complète évolutivité dans le temps et les “fonctionnalités”, ainsi qu'une réelle adéquation des solutions logicielles proposées aux besoins des utilisateurs* » (<http://www.idealx.org/dossier/oss/index.fr.html>).

I. Etat des lieux

Avec la participation du CREN et de Geo2i, nous avons tout d'abord analysé le système actuel, étape indispensable à la réalisation du prototype de saisie et de consultation des données situées sur un serveur à distance. Puis, nous avons étudié les besoins du CREN afin de mettre en place un système respectant le cahier des charges (exigences/contraintes et attentes).

A. Analyse de l'existant

1) Le matériel et les logiciels

Le Conservatoire Régional des Espaces Naturels possède six postes informatiques (quatre postes sur Toulouse et un poste pour chacun des chargés de mission travaillant chez eux). Ils disposent tous d'une connexion Internet.

En ce qui concerne les logiciels, il possède un nombre restreint de licences. Ainsi, il dispose de deux licences ESRI :

- une licence fixe ArcGIS® 9.0 sur laquelle a été développée une série d'utilitaires adaptés à leurs besoins.
- Une licence fixe ArcView® 3.2.

Ils travaillent sur le Système de Gestion de Base de Données ACCESS®.

2) Les données

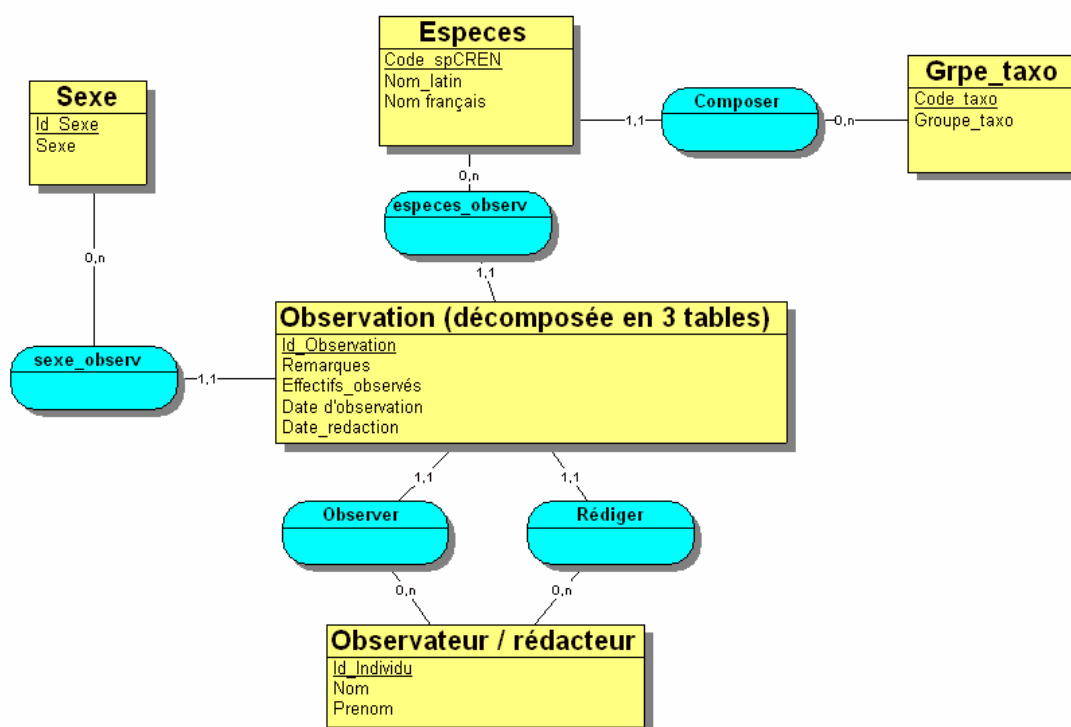
Le CREN dispose d'un grand nombre de bases de données thématiques relatives aux espaces naturels et leur biodiversité : BD¹ CREN (exemple : la base « Chiroptères »), BD Life tourbière, BD ZH (zones humides), BD ZNIEFF....

Le projet réalisé s'appuiera sur un échantillon de données issues de la base du CREN décrivant les observations de terrain relatives à la faune et à la flore effectuées par les différents intervenants. Cet échantillon servira de test pour la mise en place du prototype.

¹ Base de données

B. Modélisation des données

1) Modèle Conceptuel de Données (MCD)



Lecture du MCD :

Un groupe taxonomique est composé de aucune à plusieurs espèces (on part du principe que la table « grpe_taxo » est renseignée par des groupes qui n'ont pas forcément d'espèces répertoriées dans la base de données).

Une observation est effectuée par un observateur, qui peut (ou non) en être le rédacteur. Elle porte sur une espèce et/ou son habitat et peut concerner soit un, soit plusieurs individus. Le sexe sera indéterminé s'il s'agit d'un groupe d'individus.

2) Modèle Logique de Données (MLD)

Especes = {code_spCREN, Nom_latin, Nom_français, #Code_taxo}

Grpe_taxo = {Code_taxo, Groupe_taxo}

Observateur = {Id_individu, Nom_obs, Prenom_obs}

Sexe = {Id_sexe, Sexe}

Observation =

{Id_obs, Remarque, Effectifs_observes, date_observation, date_redaction,
#id_Observateur, #id_Rédacteur, #id_sexe, #Code_spCREN, #code_habitat}

La table « Observation » est déclinée en trois tables afin de gérer les champs géométriques, à savoir une table point, une table ligne et une table polygone.

II. Définition des besoins et des attentes

Objectif : mise en œuvre d'un outil de saisie et de consultation de données répertoriées dans le cadre d'études et d'inventaires divers visant à améliorer la connaissance du patrimoine naturel régional.

Cet outil sera accessible via Internet avec accès sécurisé.

- Il doit répondre à différents besoins permettant :
L'affichage de données raster et vecteur
La saisie de données naturalistes (géographiques et attributaires)
La consultation des données par navigation ou par requêtes simples
L'importation et l'exportation des données géographiques et attributaires des différentes bases.

- Les exigences du CREN sont détaillées ci-dessous.

A. Niveau Technique :

1) Type de logiciel

Les solutions proposées devront utiliser des technologies Open Source.

L'outil devra être utilisé par plusieurs utilisateurs excentrés du siège.

L'échange et la visualisation des données géographiques et attributaires seront effectués par l'intermédiaire d'un serveur permettant leur gestion.

2) Sécurisation des accès :

L'accès à la base devra être sécurisé par « login » et « mot de passe », afin de garantir la confidentialité des données.

3) Administration

Les solutions proposées devront prendre en compte la maintenance et l'administration des données.

B. Application

L'outil devra permettre l'affichage, la saisie, la consultation et l'import/export des différentes données.

1) Affichage des données de référence.

En vue de la saisie et de la modification des données, il doit être capable d'afficher des fonds de cartes sur lesquels seront superposées les nouvelles données.

Ces fonds sont constitués :

- d'une couche représentant les départements et les communes. Un menu permettra de sélectionner, d'afficher et de zoomer une entité géographique qui servira de support à la saisie.
- d'une couche de dalles *Scan 25*. Un menu permettra de choisir et d'afficher la sélection souhaitée.
- éventuellement d'une couche de dalles *BD Ortho* pour affinage lors de la digitalisation.

Les temps de chargement et d'affichage doivent être pris en considération.

2) Saisie des données naturalistes

Les solutions retenues devront permettre la saisie de données géographiques sous les trois formes classiques (point, ligne, polygone).

L'application devra permettre, via l'appel de formulaires, la saisie des données dans les différentes tables attributaires.

Les dates d'enregistrement et le nom du rédacteur seront intégrés automatiquement par l'intermédiaire du login saisi lors de l'accès sécurisé.

3) Consultation

La consultation se fera de deux manières :

Par navigation : pan et zoom (outil de sélection ou échelle),

Par requêtes spatiales simples (exemple : recherche d'une espèce par département et/ou par commune).

4) Imports/exports

A. Imports

Un import sera prévu afin de mettre les données déjà existantes dans la base de données. Cette opération permettra de faire une base de travail pour la visualisation des données existantes.

B. Exports

- en type de fichier SIG

Des exports devront être possibles en type de fichier SIG au format *shape* pour effectuer notamment les éditions cartographiques nécessaires pour les rapports d'étude, plan de gestion...

- vers une autre base de données

L'outil devra intégrer une procédure d'exportation des données attributaires (passerelle) vers une base de données existantes (exemple vers base de données ACCESS® pour les données chiroptères).

5) Données accessibles

Les données disponibles sont :

Les données géographiques (limites départementales et communales)

Les fichiers rasters scan 25 et la BD Ortho.

Les données déjà existantes (faune, flore, ZNIEFF, Nature 2000, ...).

C. Droit des données

1) Les données « Raster »

Dalles Scan25© de l'IGN

Dalles BD_Ortho©du département de Haute-Garonne

Il existe cinq types de licences IGN, donnant des droits différents aux licenciés :

Licence Standard : autorise le licencié à installer et à utiliser les fichiers sur le nombre de postes de travail défini par la licence, pour un usage interne. Le licencié est autorisé à représenter les fichiers ou cartes établies à partir des fichiers. Le licencié est autorisé à mettre les fichiers à disposition d'un prestataire de service pour la satisfaction de ses besoins propres, en conformité avec les droits qui lui ont été concédés et notamment dans la limite du nombre de postes de travail autorisé par la licence. Le prestataire de service est autorisé à exploiter le fichier pour les seuls besoins des prestations qui lui ont été confiées par le licencié. Le prestataire s'engage à restituer au licencié ou à détruire, à la fin de la prestation, les documents, les supports de fichiers et les fichiers mis à sa disposition.

Licence Serveur : autorise le licencié à mettre les fichiers installés sur son serveur à disposition d'utilisateurs internes ou externes, à des fins d'usage en ligne, excluant toute copie ou téléchargement des fichiers. Le licencié est responsable de l'accès des utilisateurs à son serveur, limité au nombre maximum d'utilisateurs potentiels et au nombre maximum de connexions simultanées définis par la licence. Le licencié doit prendre toutes dispositions utiles, techniques et contractuelles, pour garantir le respect par les utilisateurs des droits qui leur sont concédés. La licence serveur autorise les utilisateurs à utiliser en ligne les fichiers pour leur usage interne, sur la configuration d'exploitation décrite par la licence. La copie ou le téléchargement des fichiers, ainsi que toute utilisation hors-ligne, ne sont pas autorisés. Les utilisateurs sont autorisés à représenter les fichiers ou cartes établies à partir des fichiers

Licence de Représentation électronique : La licence de représentation électronique autorise le licencié à mettre des images numériques raster à disposition d'utilisateurs internes ou externes, à des fins de consultation, sur le support prévu par la licence (site Internet, cédérom,...). Les mentions obligatoires prévues à l'article 3 doivent figurer de façon permanente sur les images représentées. Sont exclus les traitements cartographiques, les requêtes (calcul d'itinéraires, sélection d'objets en fonction de leurs caractéristiques...)

Licence d'Evaluation et de Test : La licence d'évaluation et de test autorise le licencié, pour une durée courte définie dans la licence, à installer et à utiliser les fichiers sur le nombre de postes de travail défini par la licence, pour un usage interne, dans le but de prendre connaissance de leur contenu, de leur qualité et de leur spécification, et de tester leur adaptation aux usages du licencié. Cette licence exclut tout autre usage, et en particulier toute exploitation opérationnelle des fichiers.

Licence de Démonstration : La licence de démonstration autorise le licencié à installer et à utiliser les fichiers sur le nombre de postes de travail défini par la licence, pour un usage interne, dans le but de mettre au point et de promouvoir l'application ou le service qu'il développe et commercialise auprès de tiers. L'utilisation des fichiers est limitée aux démonstrations faites par le licencié sur des machines détenues par lui. Le licencié est autorisé à représenter les fichiers ou cartes établies à partir des fichiers, sans limitation de nombre d'exemplaires pour les documents graphiques.

Article extrait de : <http://intercarto.com/html/fr/cartotheque/ign/cgv.php>

2) Les données « Vecteur »

Les données vecteur disponibles au CREN sont issues des observations faites par les chargés de missions et techniciens. La structure en a donc l'entière propriété, d'utilisation et de cession.

Rappel de quelques définitions :

LICENCIE : personne physique, personne morale, service ou entité opérationnelle d'une personne morale, détenteur d'une licence d'utilisation de fichiers IGN.

UTILISATEUR : personne physique ou morale accédant à des fichiers ou des images numériques raster issues des fichiers, mis à sa disposition par le licencié, dans les conditions prévues par les présentes conditions générales et par la licence.

USAGE INTERNE : usage des fichiers pour la satisfaction des besoins propres du licencié, pour l'accomplissement de l'objet social ou de la mission de service public dont il est chargé. L'usage interne exclut toute exploitation commerciale des fichiers.

USAGE DOCUMENTAIRE : usage d'illustration pour localiser une information où le fond cartographique tient une place mineure et ne constitue pas l'essentiel du document, site ou cédérom.

USAGE EN LIGNE : usage où l'utilisateur accède aux fichiers par l'intermédiaire d'une connexion opérationnelle pendant toute la durée d'utilisation du fichier. La copie ou le téléchargement volontaire, même partiel, même provisoire, des fichiers, sont interdits. L'utilisation des fichiers n'est plus possible dès lors que la connexion est interrompue (usage hors-ligne). Certains logiciels réalisent automatiquement des copies du fichier sur le poste de l'utilisateur : l'usage de ces copies est interdit une fois la liaison interrompue.

FICHIER : tout fichier contenant des données géographiques numériques issues des bases de données de l'IGN, ou contenant des données géographiques numériques coproduites, coéditées et diffusées par l'IGN.

IMAGE NUMERIQUE RASTER : fichier raster (image découpée en pixels), sans information de géoréférencement, issu d'une base de données raster, du scannage d'un document, ou réalisé à partir d'une base de données vectorielles.

III. Déroulement du projet

A. Planning

Le projet a été décomposé en quatre semaines. Tous les vendredi matin, une réunion était organisée au CREN afin de faire le point sur la semaine écoulée : problèmes rencontrés, état d'avancement du projet, réorientation, ...

Notre travail peut être décomposé en quatre grandes parties :

- Un travail préalable qui comprend la familiarisation avec le sujet et les technologies Open Source
- Un ensemble d'opérations sur le serveur, à savoir l'installation du système d'exploitation et du SGBD relationnel PostGreSQL/PostGIS.
- La création de la base de données : structuration des tables sous Access afin de coller à notre Modèle Logique de Données, intégration des données attributaires sur le serveur (Access vers PostGreSQL et ArcGIS vers PostGIS).
- Le choix du logiciel SIG client : installation des logiciels, tests sur les données en attaquant le serveur, choix du logiciel et rédaction du guide d'utilisation du logiciel.

Voici comment notre projet s'est articulé dans le temps :

	Semaine 1					Semaine 2					Semaine 3					Semaine 4					
	ve	lu	ma	me	je	ve	lu	ma	me	je	ve	lu	ma	me	je	ve	lu	ma	me	je	ve
Travail préalable																					
Définition des besoins / etat des lieux																					
Modélisation des données																					
Définition du cahier de test																					
Documentation générale																					
Travail sur le serveur																					
Installation du système d'exploitation																					
Installation des bibliothèques/logiciels																					
Rédaction du cahier d'installation																					
Création de la base de données																					
Modélisation des données sous Access																					
Installation de PgAdmin																					
Transfert de la base sur le serveur																					
Choix du logiciel SIG client																					
Installation des logiciels																					
Tests sur les données																					
Rédaction des guides d'utilisation																					
Préparation de la soutenance																					
Documentation Internet																					
Rédaction du rapport																					
Installation d'un 2ème système pour présenter les tests au CREN																					

Réunions tous les vendredi matin

- Rédaction et documentation
- Installation des logiciels
- Travail sur la base de données

B. Documents à réaliser

Différents documents de suivi sont nécessaires pour le suivi du projet au CREN et sont décrits ci-dessous :

1) Guide d'installation et de maintenance du système sur le serveur

Une fois les diverses solutions envisagées, nous avons procédé à leur installation. Sachant que l'architecture du système est composée de :

- * Système d'Exploitation Open source
- * **SGBDR** (Système de Gestion de Base de Données Relationnel) coté serveur
- * Système de stockage, restitution et requêtes des données spatiales : extension spatiale du SGBDR
- * Application cartographique côté serveur

Cf. [ANNEXE 1](#)

2) Guide explicatif de création de la base de données

Un guide de création de la base de données a été élaboré. Il contient :

- La structuration des données sous Access
- L'installation du logiciel d'administration de la base de données
- La configuration de la base de données sur PostGreSQL/PostGIS

Cf. [ANNEXE 2](#)

3) Guide d'utilisation du logiciel client

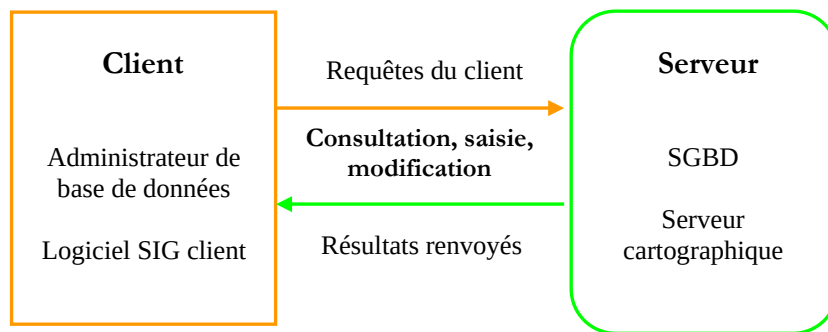
Un manuel d'utilisation du logiciel SIG choisi a été réalisé afin d'aider les chargés d'études et l'administrateur de données du CREN à se familiariser au logiciel.

Cf. [ANNEXE 3](#)

IV. Définition de la solution technique

A. Organisation du système

Nous devons définir une organisation constituée d'un poste serveur et de postes clients. L'architecture peut être schématisée de la manière suivante :



1) Le système d'exploitation

La demande principale est que le serveur soit basé sur un système d'exploitation Open Source. Le seul système d'exploitation correspondant à ce critère est Linux. Nous avons opté pour une distribution Debian Sarge 3.1 (dernière version stable), système avec une orientation assez professionnelle.

Les postes client doivent, quant à eux, rester sur un système d'exploitation Windows.

2) Le SGBD

Les données doivent être stockées sur le serveur, soit sous forme de fichiers, soit sous forme de base de données. Nous avons choisi de travailler avec un SGBD afin de structurer au maximum nos données et de rendre le système pérenne.

Il n'existe pas beaucoup de SGBD capables de gérer les données attributaires et géographiques. Nous pouvons citer MySQL/MyGIS ou encore PostgreSQL/PostGIS.

Le SGBD le plus utilisé pour la gestion des données géographiques est PostgreSQL et son extension spatiale PostGIS. Ce dernier est en pleine évolution. Il est le plus abouti dans le domaine de l'Open Source. Au contraire, l'extension MyGIS de MySQL est en cours de développement.

Nous utiliserons donc les versions PostgreSQL 8.1 et PostGIS 1.1.

L'installation du serveur Linux (avec le système d'exploitation et le SGBD) est décrite en [ANNEXE 1](#).

3) Le logiciel SIG client

Nous avons consulté le travail réalisé d'étudiants SIGMA 2005 (« Recherche de solutions applicatives Open Source autour de Mapserver » - Cf. Sources) afin de se familiariser avec les logiciels Open Source et de cibler ceux qui pourraient répondre aux attentes du CREN.

Cependant, le monde de l'Open Source est en pleine essor, de nouvelles versions sortent tous les mois, voire toutes les semaines. Il faut donc rester attentifs au quotidien sur l'évolution de ces nouvelles technologies. Nous avons donc effectué des recherches sur Internet afin de prendre connaissance des nouvelles versions des logiciels SIG développés et connaître les dernières fonctionnalités disponibles.

Les deux principaux critères sont l'affichage de données raster et la saisie et la modification de données attributaires et surtout graphiques.

L'année dernière, tous les logiciels SIG ne supportaient pas les formats de fichier vecteur et raster. Ce n'est plus le cas aujourd'hui, le critère discriminant reste donc la saisie des données.

Quatre logiciels SIG clients se démarquent des autres :

- **OpenJUMP** développé par Vivid Solution.
- **Quantum Gis** réalisé par un grand nombre de développeurs.
- **Udig** développé par Refractions.
- **Thuban** développé par l'équipe Intevation.
- **Grass** développé par un grand nombre d'universités et d'entreprises.

Le choix de la meilleure solution est réalisé d'après les tests effectués sur les machines.

Grass

Avantages

Il supporte un grand nombre de format de fichiers vecteur et raster. C'est un logiciel très puissant qui offre un grand nombre de fonctionnalités SIG, notamment la saisie de données graphiques.

Inconvénients

Grass n'est pas facile à installer et nécessite 2 Gigas d'espace disque disponible. De plus, il est lourd à utiliser et n'est pas du tout intuitif.

Thuban

Avantages

Il peut lire des données vecteurs/raster, offre la possibilité de réaliser des jointures et des requêtes et permet d'exporter en format vectoriel.

Inconvénients

Ce logiciel est difficile à installer notamment dû à la complexité des bibliothèques Python.

En effet, Thuban est codé en Python et il est indispensable d'installer le logiciel qui permet d'interpréter ce langage.

De plus, il ne permet pas beaucoup de choses : aucune mise à jour ni suppression des données graphiques, l'export est limité, ...

Udig

Avantages

Il est développé par le même groupe que PostGIS, la connexion à la base de données ne pose donc pas de problèmes.

Il permet l'édition et la modification de la géométrie en ligne en temps réel, la connexion aux bases de données Oracle Spatial ou PostGIS.

Inconvénients

Il présente quelques bugs d'affichage des objets géographiques, par l'exemple lors de la création de polygones. Même l'actualisation de la page (« Rafraîchir ») ne permet pas l'affichage de la nouvelle entité saisie. Nous préférons donc le laisser de côté.

De plus, il propose très peu de fonctionnalités SIG (seulement une requête de sélection basique).

Quantum Gis

Avantages

Ce logiciel est facile à installer, son interface est très conviviale et proche de celle d'ArcView. En effet, il fonctionne sous la forme de projet tout comme ce SIG propriétaire.

Il se connecte directement à PostGIS et permet d'extraire toutes ou une partie des données stockées sur ce SGBD.

De plus, nous pouvons exporter en Mapfiles, mais aussi de les envoyer sur le serveur PostGreSQL/PostGIS. Cela peut être intéressant pour le CREN dans la mesure où il envisage, plus tard, d'utiliser un serveur cartographique.

Inconvénients

Il ne gère pas les champs au format « date » et ne permet pas à l'heure actuelle la modification des données graphiques.

OpenJUMP

Avantages

Ce logiciel propose une interface graphique conviviale avec beaucoup de fonctionnalités : analyses thématiques, géotraitements (réalisation de buffers notamment), requêtes spatiales et attributaires.

C'est un des seuls logiciels proposant la création de données, mais aussi la modification et la suppression des objets.

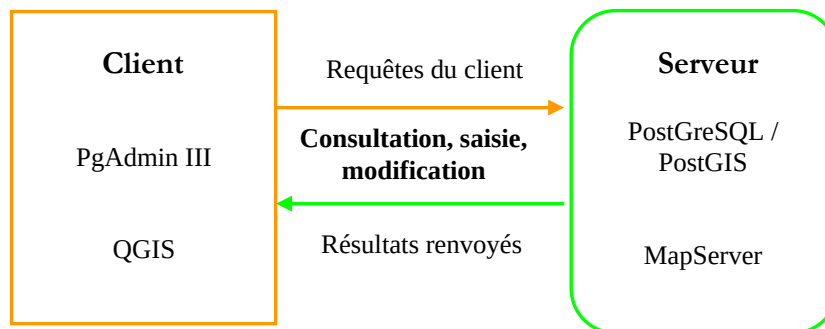
De plus, il est possible d'exporter au format shape.

Inconvénients

La version 2.0 de OpenJUMP ne permet pas de travailler avec un PostGreSQL version 8.1 couplé à un PostGIS version 1.1, configuration installée sur notre serveur. OpenJUMP ne fonctionne actuellement qu'avec une version 7.4 de PostGreSQL. Nous n'avons pas testé cette solution car il fallait compiler les sources afin d'installer le SGBD sur un système Windows.

La sortie d'une nouvelle version d'OpenJUMP permettant de s'y connecter est prévue pour la semaine du 20 février 2006.

Schéma général de l'application choisie :



Nous préconisons donc l'utilisation du logiciel Quantum GIS pour l'instant. Il ne permet pas la modification de données graphiques, mais est le seul logiciel SIG libre stable qui gère très bien la connexion avec PostGreSQL/PostGIS. Dans l'avenir, il faudra peut-être revoir cette organisation si les problèmes et les limites rencontrés dans les autres SIG sont résolus.

4) L'organisation choisie répond-elle aux attentes du CREN ?

La gestion des données :

L'affichage des données vecteur et raster est respectée.

En ce qui concerne les données raster, elles ne sont pas gérées par PostGIS. Il faudra donc les stocker sur le serveur ou sur les postes clients. Le fait de stocker ces données sur le poste client en local diminuera le temps d'affichage des couches par rapport au serveur.

La saisie de données géographiques et attributaires :

L'édition (création/modification/suppression) de données attributaires ne pose aucun problème.

En ce qui concerne les données graphiques, il est possible de créer et de supprimer des objets. Par contre, leur modification n'est pas rendue possible par tous les logiciels. En effet, QGIS ne permet pas de modification de la géométrie.

Au contraire, celle-ci est proposée par OpenJUMP et Udig, mais n'est pas toujours stable.

La consultation des données par requête et navigation :

La consultation des données est simple sur tous les logiciels SIG retenus. Pour ce qui est des requêtes, QGIS propose de simples requêtes de sélection (tout comme Udig).

Par contre, des requêtes beaucoup plus complexes sont présentes sur d'autres logiciels comme OpenJUMP.

Importation des données :

Deux passerelles ont été créées pour transférer les données d'Access et ArcView sur le SGBD PostGreSQL/PostGIS :

- pour les données attributaires : une liaison ODBC reliant Access à PostGreSQL en local sur Windows, puis un transfert des données sur PostGreSQL du poste client vers le serveur.

- Pour les données graphiques : le lancement de l'application de GDAL « shp2pgsql.exe » dans l'invite de commande.

Exportation des données :

Les deux mêmes passerelles ont été testées dans le sens inverse :

- pour les données attributaires : une connexion ODBC
- pour les données graphiques : le lancement de l'application de GDAL « pgsql2shp.exe » dans l'invite de commande

B. Les perspectives du projet

1) Évolution des choix techniques

Du fait de l'évolution rapide du monde de l'Open Source, il se peut que de nouvelles versions, voire de nouveaux logiciels SIG, apparaissent.

Ainsi faudra-t-il surveiller l'actualisation des logiciels que nous avons sélectionnés et testés, notamment :

- La nouvelle version d'OpenJUMP (d'ici quelques jours) qui doit pouvoir se connecter à une base PostGIS 1.1. Si la création et la modification des données sont effectives dans la base de données sur le serveur, c'est à notre avis, la solution logicielle répondant le mieux aux besoins du CREN.
- Udig est intéressant mais présente de gros bug d'affichage et peu de fonctionnalités. Mais une évolution de ce logiciel pourrait le rendre compatible aux attentes du CREN.

D'autres options peuvent être envisagées :

- Le logiciel GvSIG qui est actuellement un simple viewer mais qui prévoit des nouvelles fonctionnalités de saisie et de modification des données graphiques.
- La nouvelle version de CartoWeb 3.2 (février 2006) permet de faire de la saisie et de la modification de données. Mais l'installation est longue et compliquée. De plus, elle se fait sur le serveur.

2) Évolution du projet

Le projet est amené à évoluer notamment au sujet des données traitées.

Tout d'abord, les données concernant les habitats observés n'ont pas été intégrées dans le prototype. En effet, les chargés de mission du CREN devaient redéfinir la structure des informations liées aux habitats (par exemple le type de géométrie : ponctuel et/ou linéaire et/o surfacique).

De plus, les données liées aux tourbières ont aussi été mises de côté.

V. Bilan

A. Les problèmes rencontrés

Nous avons rencontré un certain nombre de **problèmes techniques** :

- L'installation du serveur sur système d'exploitation Linux à cause de notre manque de connaissances et d'expériences
- L'installation des versions d'Apache et de PHP ainsi que des différentes bibliothèques nécessaires aux logiciels (serveur Linux).
- L'intégration des données sous PostGreSQL/PostGIS (problème de clés étrangères dues au jeu de données).
- L'installation de quelques logiciels SIG a posée problème (notamment Grass, Thuban, Jump)

Quelques **problèmes organisationnels** ont été rencontrés :

- L'éloignement de l'encadrant technique à Aix-en-Provence.
- Le blocage de l'Université du Mirail pendant quelques jours.

B. De nouveaux acquis techniques et pratiques

Nous disposons maintenant d'un guide technique qui nous permettra de reproduire l'ensemble des manipulations effectuées. Même si les logiciels utilisés seront amenés à évoluer, nous avons acquis ce mois-ci la connaissance des principes de fonctionnement d'un tel système.

Le fait de réaliser un projet dans son intégralité et sur le moyen terme est un exercice que nos diverses formations ont trop souvent négligé. Ainsi, ce mini projet nous apporte une première expérience concrète en la matière.

L'ensemble des recherches effectuées nous a permis de nous mettre au fait des capacités concrètes des technologies « Open Source » actuellement en plein essor dans le domaine des SIG. Leur connaissance nous sera d'une grande utilité aussi bien dans notre stage que dans notre futur métier de géomaticien.

Un point très positif pour l'ensemble du groupe a été l'initiation au système d'exploitation LINUX, couramment utilisé dans les entreprises et concurrent le plus sérieux de Windows. Nous avons pu appréhender, grâce à Laurent JEGOU, l'approche particulière de LINUX (systèmes de fichiers par package, utilisation de l'invite de commande, compilation des sources, ...)

Enfin, ce projet aborde deux aspects de l'évolution future des SIG :

- d'une part les technologies Open Source
- d'autre part les architectures client/serveur qui garantissent leur utilisation à distance.

Sources

Sites Internet et forums sur Linux :

- Sites généralistes
<http://www.lea-linux.org>
<http://linuxfr.org>
- Documentation pour l'installation de MapServer et de tous les programmes et bibliothèques nécessaires sur une distribution Linux
<http://mapserver.gis.umn.edu/docs/howto/verboselinuxinstall>

Sites Internet et forums sur les technologies Open Source :

Généralistes :

- Forum sur les SIG
<http://forumsig.org>
- Site de développement de logiciels Open Source
<http://sourceforge.net>
- Site sur les SIG libres :
<http://sig-libre.org>

PostgreSQL :

- Site des utilisateurs de PostgreSQL
<http://www.postgresql.org>

PostGIS :

- Site Internet de la communauté des utilisateurs de PostGIS francophone
<http://postgis.fr>
- Installation de PostgreSQL/PostGIS (site de Jean-David TECHER)
<http://www.01map.com>
- Articles et tutoriaux sur PostGIS (site de Jean-David TECHER)
<http://techer.pascal.free.fr/>

Logiciels SIG :

- Site référence de Quantum GIS
<http://qgis.org>
- Projet-SIGLE (Systèmes d'Information Géographique Libre) sur les logiciels SIG libres, et en particulier sur OpenJUMP et GvSIG
<http://www.projet-sigle.org>
- Site de Thuban
<http://thuban.intevation.org>
- Site du logiciel Udig
<http://udig.refrations.net/confluence/display/UDIG/Home>

Sites Internet sur droit des données :

- Site du CNIG
<http://cnig.les-argonautes.fr/>
- Site de la CNIL
<http://www.cnil.fr/>
- Site présentant le travail d'étudiants du Mastère SILAT sur la diffusion de données liée à l'information géographique
<http://sig.ish-lyon.cnrs.fr>
- Site de la société Intercarto avec une rubrique consacrée aux droits des données
<http://intercarto.com/html/fr/cartotheque/ign/cgv.php>

Rapports et documents :

- BLANC Samuel, COUTARD Marie, KLIPFEL Jean-Pascal : « *Recherche de solutions applicatives Open Source autour de MapServer* », Projet DESS Sigma, Février 2005.

<http://sigfrance.free.fr/ressources/filebrowser/downloads/MapServer/AutourDeMapserver.pdf>

ANNEXES

ANNEXE 1

GUIDE D'INSTALLATION ET DE MAINTENANCE

Sommaire

I.	INSTALLATION DU SERVEUR LINUX	24
A.	INSTALLATION DU CD 1 BOOTABLE	24
B.	CONFIGURATION DU SYSTEME	25
C.	LECTURE DES AUTRES CD	25
D.	AUTRES CONFIGURATIONS SUR LE SERVEUR LINUX	26
1)	<i>Configuration de recherche des paquets</i>	27
2)	<i>Mise à jour de la version de PHP et d'Apache</i>	27
II.	INSTALLATION DE POSTGRESQL/POSTGIS	35
A.	POSTGRESQL	35
1)	<i>Installation avec les fichiers binaires</i>	35
2)	<i>Installation avec des codes source</i>	36
B.	POSTGIS	41
1)	<i>Installation de Geos</i>	41
2)	<i>Installation de Proj 4</i>	42
3)	<i>Installation de PostGIS</i>	44
III.	MAPSERVER	44
A.	INSTALLATION DE GDAL	44
B.	INSTALLATION DE MAPSERVER	45
	QUELQUES COMMANDES UTILES	49

I. Installation du serveur Linux

Linux est composé d'un noyau (kernel) sur lequel se greffent différentes applications. C'est un petit logiciel qui reconnaît le matériel et qui permet de gérer les fichiers et les applications. Une distribution, c'est tout ce qui se greffe autour de ce noyau, à savoir un regroupement de logiciels. A l'origine, elle s'installait avec une disquette, puis avec des cédéroms.

Au départ (il y a 20 à 25 ans), les utilisateurs compilaient les codes sources qu'ils installaient sur leur noyau Linux. La compilation se faisait manuellement, à la volée. Afin de structurer tout cela, l'organisation des distributions se fait par paquets (logiciels compressés qui contiennent tous les paramètres de configuration et d'installation).

Il existe deux grands types de distributions Linux :

- Debian (professionnelle, sécurisée)
- Red Hat (système de paquets RPM – utilisé par quasiment toutes les autres distributions Linux, comme Debian).

Nous optons pour une Debian 3.1, nom de code Sarge, la dernière version stable de Debian (datant de 2002).

Cette distribution est constituée de 6 CD.

A. Installation du CD 1 bootable

- Lancement de l'installation
- Définition de plusieurs paramètres : la langue, le pays, le clavier (français à chaque fois).
- Détection du matériel, du réseau, ...
- Configuration du réseau. Elle permet d'accéder au réseau et de faire les mises à jour du système Linux (et en particulier les mises à jour de sécurité). L'installation propose plusieurs méthodes de configuration :
 - o Configuration DHCP (paramétrage automatique du réseau)
 - o Configuration par soi-même
 - o Pas de configuration du réseau pour le moment.

C'est cette dernière solution que nous choisissons car nous n'avons pas le temps de réaliser toutes les mises à jour. Il faudra de toute manière revenir à cette étape plus tard.

- Demande de donner un nom au réseau (ou DSN). Ces caractères sont liés à l'adresse IP de la machine. Ainsi, pour appeler une machine en mode client/serveur, on peut soit utiliser son nom de réseau (CF. ci-dessus), soit utiliser son numéro d'IP.

Ici, nous l'appelons « sigma S2 ».

- Partitionnement du disque dur.

Sur Linux, nous avons besoin de deux partitions :

- o Partition globale. L'installation nous propose d'allouer 19,2 GB pour la racine en EXT3.
- o Partition particulière pour le SWAP. C'est un fichier utilisé lorsque toute la mémoire vive est utilisée. Un morceau du disque dur est consacré à ce fichier SWAP.

Sous Linux, le système de fichiers NTFS n'existe pas. On peut lire des périphériques en FAT 32 ou en NTFS (et écrire en FAT 32), en installant une sorte de driver. Linux reconnaît un assez grand nombre de systèmes de fichiers (propres à Linux) comme EXT 2, EXT 3, SWAP, ... L'installation nous propose d'allouer 855 MB pour la partition SWAP.

- Configuration des paquets de base. On choisira plus tard les autres paquets à installer.
- Lancement de GRUB : programme de démarrage de Debian. Il détecte s'il existe un autre système d'exploitation sur la machine. Si c'est le cas, il proposera à l'ouverture de l'ordinateur de choisir entre les deux systèmes.

La première phase d'installation est terminée et le noyau dur de Debian est installé. On redémarre l'ordinateur et on enlève le CD dans le lecteur.

B. Configuration du système

- Configuration de l'heure
- Configuration du fuseau horaire
- Configuration des utilisateurs. Au départ, Linux était fait pour gérer des données et du matériel. La notion de sécurité est donc très importante. C'est pourquoi la création de comptes utilisateur avec identification est indispensable. Sur Linux, chaque fichier contient les droits d'utilisation des utilisateurs ou des groupes d'utilisateurs.
 - o Paramètres du super-utilisateur (root ou administrateur) : il a tous les droits sur le serveur. Nous choisissons comme mot de passe « sigma », qu'il faut ressaisir une deuxième fois pour confirmation.
 - o Création d'un premier compte utilisateur. Nous choisissons pour login et pour mot de passe « sigma ».Nous pourrions créer d'autres comptes plus tard, après l'installation.
- Faut-il installer le système via une connexion PPP ? Non
- Mise en œuvre de « Apt » : outil de gestion des paquets sous Debian. Nous avons le choix entre plusieurs méthodes d'accès : cédérom, ftp, http, ... Nous choisissons « cédérom » car nous avons tous les paquets qui nous intéressent sur les CD.

C. Lecture des autres CD

- On insère le deuxième CD. Pour lancer le CD, il faut taper les commandes « /dev/cdrom ».
- L'outil « Apt » lit sur le CD un fichier qui liste tous les paquets présents. Il faut insérer tous les autres CD pour que Apt détermine tous les paquets disponibles pour l'installation.

Après la lecture de tous les CD, Apt fait une liste qui regroupe tous les logiciels et les regroupe par « paquet » (ou thématique). Ceci est d'ailleurs beaucoup plus développé sur une distribution Red Hat.

Régulièrement, nous avons des messages qui indiquent que les mises à jour de sécurité n'ont pas pu être réalisées. Tout ceci est normal dans la mesure où nous n'avons pas configuré la connexion réseau.

- Lancement de l'installation des paquets. Voici les paquets proposés :
 - o Environnement graphique de bureau (à sélectionner)

- o Serveur Web (à sélectionner)
- o Serveur d'impression
- o Serveur DNS (gestion des domaines réseau)
- o Serveur de fichiers
- o Serveur de courrier
- o Base de données SQL (MySql, PostGreSQL)
- o Choix manuel des paquets et installation personnalisée.

Nous n'installons que les deux premiers paquets. En effet, quand l'environnement graphique sera paramétré, nous pourrons installer d'autres paquets.

De plus, comme cette distribution date de 2002, les logiciels comme PostGreSQL ne sont plus à jour, nous irons donc les chercher sur Internet.

- Configuration de Exim 4 : nous la laissons de côté, cela correspond à la configuration des mails.
- Configuration de ssh : protocole sécurisé. Le programme demande quelle version nous souhaitons installer : faut-il autoriser la version 2 ? nous disons « oui » à chaque fois.
- Voulez-vous utiliser le serveur shd ? Non

A chaque fois que nous ne savons pas ou que la question ne nous concerne pas, nous gardons les paramètres par défaut.

- Configuration du matériel :
 - o Les cartes vidéo, son et mère : on choisit la détection automatique du matériel
 - o Le serveur x (= xwindows) : permet de gérer les fenêtres d'autres machines en se connectant dessus grâce au système de client/serveur sous Linux.
 - o La souris : détection automatique.
On doit choisir le port de branchement de la souris.
Pour un branchement en USB : « /dev/input/mice »
Pour un branchement PS/2 : « /dev/psaux »
 - o L'écran : détection automatique. Nous devons choisir s'il s'agit d'un écran à cristaux liquides ou un CRT.
Nous devons définir les valeurs de rafraîchissement horizontal de l'écran car cela n'est pas détecté automatiquement par le programme. Pour cela, il faut regarder la documentation de l'écran.
 - o Choix de la profondeur de couleur (16, 24 bits, ...)

On passe ensuite à la suite de l'installation.

On termine la configuration du serveur en redémarrant la machine en tapant « reboot » en lignes de commande.

D. Autres configurations sur le serveur Linux

Nous avons choisis de mettre les code source avant compilation des différentes applications et bibliothèques dans le répertoire src :

Nous passons en mode administrateur dans un terminal.

```
su root
```

Nous devons entrer le mot de passe.

Puis on tape les lignes de commandes suivantes :

```
root@sigmas2:/home/sigma# cd ..
root@sigmas2:/home# cd ..
root@sigmas2:/# cd usr
root@sigmas2:/usr# mkdir src
root@sigmas2:/usr# cd src
```

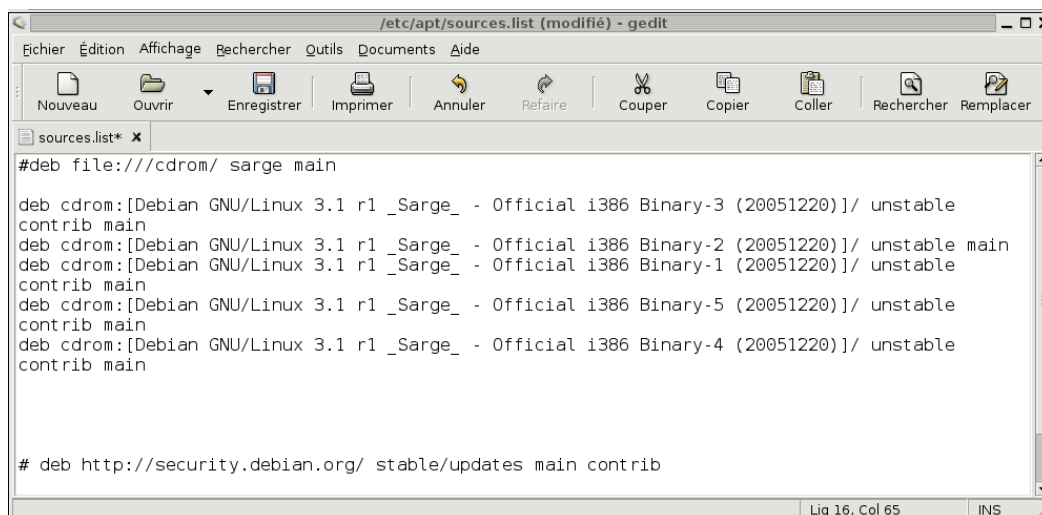
1) Configuration de recherche des paquets

Le fichier « /etc/apt/sources.list » contient les configurations de gestion des paquets. Sous Linux, il est possible de rechercher des fichiers sources sur internet en lançant une recherche « aptitude search ». Pour cela, il faut configurer les sites Internet sources. C'est ce que nous allons réaliser à présent.

Nous allons chercher les sites miroirs officiels sur le site de Debian.

Nous rajoutons ce site dans le fichier source :

```
root@sigmas2:/etc/apt/ # source.list
```



Il faut donc ajouter la ligne suivante dans le fichier « sources.list »:

```
deb http://ftp.fr.debian.org/debian sarge main contrib
#non-free
```

2) Mise à jour de la version de PHP et d'Apache

L'installation de PHP et d'Apache par paquet Debian ne permet pas d'installer les dernières versions de PHP Mapscript.

Dans le gestionnaire de paquets, nous pouvons voir que les versions PHP 4 et Apache 2 sont installées. Nous choisissons donc de les désinstaller via le gestionnaire de paquets.

a. Installation de gd2

Gd2 est une bibliothèque graphique qui permet de gérer des images avec PHP.

Recherche de la bibliothèque :

```
root@sigmas2:/usr/src# aptitude search libgd2
```

```
p    libgd2                      - GD Graphics Library
version 2
p    libgd2-dev                  - GD Graphics Library
version 2 (development version)
i A  libgd2-noxpm                 - GD Graphics Library
version 2 (without XPM support)
iBA libgd2-noxpm-dev              - GD Graphics Library
version 2 (development version)
p    libgd2-xpm                  - GD Graphics Library
version 2
p    libgd2-xpm-dev              - GD Graphics Library
version 2 (development version)
```

Rq :

« p » : la bibliothèque n'est pas installée

« i » : la bibliothèque est installée

« v » : la bibliothèque est installée virtuellement

Nous installons les bibliothèques « libgd2 » et « libgd2-dev ».

Installation des fichiers :

```
root@sigmas2:/usr/src# aptitude install libgd2
```

Nous validons lorsque l'ordinateur nous demande de supprimer quelques paquets pour en installer d'autres.

```
root@sigmas2:/usr/src# aptitude install libgd2-dev
```

b. Installation de curl

Curl est une bibliothèque qui permet de gérer les FTP.

Recherche de la bibliothèque :

```
root@sigmas2:/usr/src# aptitude search curl
```

```

p    curl                                - Get a file
from an HTTP, HTTPS, FTP or GOPHER ser
v    curl-ssl                            -
v    libcurl-dev                         -
v    libcurl-easy-perl                  -
p    libcurl-ocaml                       - ocaml curl
bindings
p    libcurl-ocaml-dev                   - ocaml
libcurl bindings
v    libcurl-ssl-dev                     -
v    libcurl2                            -
v    libcurl2-dev                        -
i A libcurl3                             - Multi-
protocol file transfer library, now with SS
p    libcurl3-dbg                         - libcurl
compiled with debug symbols
p    libcurl3-dev                         - Development
files and documentation for libcurl
p    libcurl3-gssapi                     - libcurl
compiled with GSSAPI support
p    libwww-curl-perl                     - Perl
bindings to libcurl
v    php4-cgi-curl                        -
p    php4-curl                           - CURL module
for php4
p    python2.2-pycurl                     - Python
bindings to libcurl
p    python2.3-pycurl                     - Python
bindings to libcurl
p    python2.4-pycurl                     - Python
bindings to libcurl
p    tclcurl                             - Tcl
bindings to libcurl

```

Nous installons les bibliothèques « curl » et « libcurl3-dev » (car il n'y a pas de fichier « curl-dev »).

Installation des fichiers :

```
root@sigmas2:/usr/src# aptitude install curl
```

```
root@sigmas2:/usr/src# aptitude install libcurl3-dev
```

c. Installation de Apache

Il existe deux versions de Apache : la version 1.3 et la 2.

Nous nous rendons sur le site d'Apache <http://httpd.apache.org/download.cgi> et choisissons la version 2.2 compressée en bz2.

Décompression des sources dans ce répertoire. Pour cela, nous devons utiliser le logiciel tar :

```
root@sigmas2:/usr/src# tar -jxvf httpd-2.2.0.tar.bz2
```

Rq :

« j » indique que le format de compression est bz2. Pour décompresser un fichier au format gz, nous devons mettre « z » à la place du « j ».

« x » décompresse le fichier en respectant l'arborescence des fichiers

« v » indique que la décompression est verbeuse (explique toutes les étapes réalisées)

« f » est suivi du nom du fichier à décompresser

Installation d'Apache :

Nous utilisons une page Internet sur le site de **Mapserver** qui décrit les manipulations à réaliser (CF. Bibliographie).

```
root@sigmas2:/usr/src/httpd-2.2.0# ./configure
-- prefix = /usr/local/apache2 (indique le répertoire
d'installation)
-- enable-deflate
-- enable-info
-- enable-mime-magic
-- enable-rewrite
-- enable-so (permet de gérer une bibliothèque externe
dont PHP et MapScript)
-- enable-speling
-- enable-unique-id
-- enable-usertrack
-- with-mpm = prefork
```

Nous enlevons l'installation de la bibliothèque « ssl » utilisée pour les connexions sécurisées.

Compilation et installation des sources :

```
root@sigmas2:/usr/src/httpd-2.2.0# make
```

```
root@sigmas2:/usr/src/httpd-2.2.0# make install
```

Nous lançons enfin Apache afin de vérifier qu'il fonctionne bien :

```
root@sigmas2:/usr/local/apache2/bin# apachectl start
```

Configuration du script de démarrage d'Apache : « init.d ». Celui-ci se trouve dans le répertoire « /usr/etc/ » :

```
root@sigmas2:/usr/src/httpd-2.2.0# cp
/usr/local/apache2/bin/apachectl
/etc/init.d/.
```

Nous ouvrons ce fichier de démarrage qui se trouve dans /etc/init.d/apachectl et nous copions les quatre lignes en gras de ce code :

```
#!/bin/sh
#
# chkconfig: - 85 15
# description: Apache is a Web server used to serve HTML and CGI.
# processname: httpd
# pidfile: /usr/local/apache2/logs/httpd.pid
#
# Copyright 2000 - 2005 The Apache Software Foundation or its
# licensors, as
# applicable.
.
.
.
```

Rq : le fichier ne peut se modifier qu'en mode root. A partir de la console, se mettre en root (su root, puis mot de passe), puis lancer l'éditeur de texte (ex : Abiword), afin de pouvoir modifier le fichier. Pendant l'utilisation d'abiword, si une action est effectuée dans le terminal, l'application de traitement de texte sera fermé

Modification de la racine par défaut :

```
DocumentRoot "/var/www"
```

Si l'on souhaite avoir une installation propre, il faut modifier des adresses en plus, à savoir un nom d'administrateur, un nom de domaine, ...).

Modification identique dans la balise directory :

```
<Directory "/var/www">
```

Rajout d'un répertoire temporaire en fin du fichier de configuration :

```
Alias /tmp/ /var/www/tmp/
```

Création physique de ce fichier dans le répertoire « /var/www/tmp »

Nous donnons à ce dossier tous les droits d'écriture.

d. Installation de PHP

Nous recherchons les sources de PHP sur Internet sur le site <http://www.php.net>.

Il existe deux versions : PHP 4 et PHP 5. Nous choisissons d'installer la version la plus récente.

Décompression des sources :

```
root@sigmas2:/usr/src# tar -jxvf php-5.1.2.tar.bz2
```

Installation des fichiers :

```
root@sigmas2:/usr/src/php-5.1.2# ./configure
-- prefix = /usr/local/apache2/php
-- with-apxs2 = /usr/local/apache2/bin/apxs
-- with-config-file-path = /usr/local/apache2/php (met le
fichier php.ini dans ce répertoire)
```

```
-- with-gd
-- with-pgsql
-- with-gettext
-- with-curl
-- with-zlib
-- enable-dbase (permet de lire des fichiers dbase)
```

Compilation et installation des sources :

```
root@sigmas2:/usr/src/php-5.1.2# make
```

```
root@sigmas2:/usr/src/php-5.1.2# make install
```

Poursuite de la configuration :

Nous copions ensuite le fichier « php.ini-recommended ».

```
root@sigmas2:/usr/src/php-5.1.2#
cp php.ini-recommended /usr/local/apache2/php/php.ini
```

Rq : la commande cp permet la copie, « php.ini » correspond au nouveau nom de fichier

Nous devons apporter des modifications au fichier de configuration d'Apache (/usr/local/apache2/conf/httpd.conf). Parfois, celui-ci se modifie automatiquement et nous n'avons rien à rajouter.

Voici les lignes à écrire dans le fichier :

```
Autour de la ligne 53:  
LoadModule php5_module          modules/libphp5.so  
  
Autour de la ligne 165:  
<IfModule dir_module>  
    DirectoryIndex index.html index.php index.php3  
</IfModule>  
  
Autour de la ligne 307:  
(Dans le module AddType application)  
AddType application/x-httpd-php  .php      .phtml      .php3  
AddType application/x-httpd-php-source .phps
```

Rq: ce fichier doit être modifié en mode root, lancement du programme d'édition de texte à partir d'un terminal.

Ensuite, nous relançons Apache :

```
root@sigmas2:/usr/local/apache2/bin# apachectl stop  
root@sigmas2:/usr/local/apache2/bin# apachectl start
```

On effectue ensuite un test pour vérifier que PHP fonctionnent bien : nous créons un fichier « test.php » que l'on enregistre dans le dossier "/var/www" (correspondant à la racine de localhost) qui contient les informations suivantes :

```
<? php  
  
php.info();  
  
?>
```

Une page indiquant les propriétés de PHP sont indiqués.

e. Démarrage automatique d'Apache au lancement de Linux

Pour un lancement automatique on prendra le fichier **apachectl** se trouvant sous **/usr/local/apache/bin** et on le placera sous **/etc/rc.d/init.d**.

```
root@sigmas2:cp /usr/local/apache/bin/apachectl  
  
/etc/rc.d/init.d/apachectl
```

Nous rendons le fichier « /etc/rc.d/init.d/apachectl » exécutable à partir de la commande :

```
root@sigmas2:/etc/rc.d/# chmod 777 init.d -R
```

Dans un terminal en mode root on tape, afin de lancer apache au démarrage, la commande suivante :

```
root@sigmas2:/etc/rc.d/init.d/# chkconfig apachectl on
```

Pour un arrêt de cette commande, on tape :

```
root@sigmas2:/etc/rc.d/init.d/# chkconfig apachectl off
```

Pour lancer le serveur, il suffit maintenant de taper:

```
root@sigmas2:/etc/rc.d/init.d/# /etc/rc.d/init.d/apachectl  
start
```

II. Installation de PostGreSQL/PostGIS

A. *PostGreSQL*

PostGreSQL est disponible sous différentes formes :

- Le code source à compiler : nous sommes sûrs que le logiciel fonctionnera sur n'importe quelle type de machine.
- Les fichiers binaires : ces fichiers sont déjà compilés. Il y a autant de fichiers binaires que de systèmes d'exploitation. Il faut donc choisir les fichiers correspondant à notre installation.

1) Installation avec les fichiers binaires

Pour ce qui nous concerne, nous allons utiliser des fichiers RPM (Red Hat) sur notre Debian. Nous ouvrons un terminal.

Nous passons en mode administrateur.

```
su root
```

Nous devons entrer le mot de passe.

Nous utilisons la commande « apt » pour installer la gestion des paquets RPM.

Il existe deux commandes « apt » :

- « apt-get » : permet d'installer et supprimer les applications. Mais on ne peut pas faire de recherche par mot clé.
- « aptitude » : possède le paramètre « search »

Recherche des fichiers rpm :

```
root@sigmas2:/usr/src# aptitude search rpm
```

Installation des fichiers rpm :

```
root@sigmas2:/usr/src# apt-get install rpm
```

On valide par « oui ».

Les répertoires sont vides concernant les fichiers Red Hat. Nous pourrions utiliser les fichiers pour Fedora, mais il risque d'y avoir des conflits.

Nous décidons donc d'abandonner cette procédure pour installer les codes sources directement.

2) Installation avec des codes source

Il y a 5 étapes dans l'installation du code source :

1. Récupération des sources
2. Décompression
3. Adaptation et configuration
4. Compilation
5. Installation

a. Récupération des sources

Téléchargement des codes sources.

Le code source est disponible sur le site <http://www.postgresql.org/ftp/source/v8.1.2/>

Nous sélectionnons PostGreSQL version 8.1.2. Il existe deux formats de compression : gz et bz2 (algorithmes de compression). Bz2 est actuellement le meilleur format de compression. Nous choisissons le répertoire PostGreSQL_8.1.2_bz2.

b. Décompression

Affichage des dossiers présent dans le répertoire src :

```
root@sigmas2:/usr/src# dir
```

Copie des sources dans le répertoire src :

```
root@sigmas2:/usr/src# cp /home/sigma/postgresql-8.1.2.tar.bz2 .
```

Rq : le “.” de la fin signifie que l'on copie dans le répertoire courant)

Décompression des sources dans ce répertoire :

```
root@sigmas2:/usr/src# tar -jxvf postgresql-8.1.2.tar.bz2
```

c. Adaptation et configuration

Nous utilisons un script (fichier bash) pour connaître les paramètres du système d'exploitation (test des matériels et logiciels).

Dans le répertoire qui contient le script « configure »

```
root@sigmas2:/usr/src/postgresql-8.1.2# ./configure
```

Il est possible que des messages d'erreur s'affiche en fonction des bibliothèques de fonction ne sont pas installées.

Par exemple, il indique que « readline » est manquante. Quand nous allons dans la gestion des paquets, nous pouvons voir qu'elle est déjà installée en tant que bibliothèque, mais pas en tant qu'outil de développement. Il faut donc installer le paquet « readline.dev ».

Le même message d'erreur apparaît pour « zlib ». Nous réalisons donc la même opération.

Nous notons la création d'un « makefile » (fichier de compilation).

d. Compilation des sources

La compilation se lance avec les commandes « make » ou « gmake ».

Nous sommes toujours dans le répertoire de travail.

Lancement de la commande :

```
root@sigmas2:/usr/src/postgresql-8.1.2# make
```

e. Installation des sources

Copie des fichiers exécutables sur la machine, à leur endroit de destination :

```
root@sigmas2:/usr/src/postgresql-8.1.2# make install
```

f. Autres opérations

Ajout d'un utilisateur dans le système :

```
root@sigmas2:/usr/src/postgresql-8.1.2# adduser
```

Il est proposé de répondre aux valeurs suivantes, nous répondons par les valeurs par défaut :

```
Modification des informations relatives à l'utilisateur
postgres
Entrez la nouvelle valeur ou « Entrée » pour conserver la
valeur proposée
    Nom complet []:
    No de bureau []:
    Téléphone professionnel []:
    Téléphone personnel []:
    Autre []:
Ces informations sont-elles correctes ? [o/N] 0 « adduser
»
```

On met le nom postgres et mot de passe postgres (x2)

On fait « entrée » pour les informations relatives à l'utilisateur postgres
Nom complet

Création d'un répertoire qui va stocker les données dans « /usr/local/pgsql/data » :

```
root@sigmas2: mkdir /usr/local/pgsql/data
```

Changement du propriétaire du fichier :

```
root@sigmas2:/usr/src/postgresql-8.1.2# chown postgres  
/usr/local/pgsql/data/
```

Nom du fichier

Nom du nouveau
propriétaire

Identification en tant que l'utilisateur de postgres :

```
root@sigmas2: su postgres
```

Initialisation de postgres avec le répertoire de données juste après :

```
postgres@sigmas2: /usr/local/pgsql/bin/initdb -D  
/usr/local/pgsql/data
```

Configuration du fichier log. Nous rajoutons le fichier log à la suite du fichier existant avec la commande « 2>&1 & ». On démarre la base de données.

```
postgres@sigmas2: /usr/local/pgsql/bin/pg_ctl start -D  
/usr/local/pgsql/data
```

On crée une base de données test :

```
postgres@sigmas2: /usr/local/pgsql/bin/createdb test
```

On ouvre la base de données test :

```
postgres@sigmas2: /usr/local/pgsql/bin/psql test
```

Configuration d'accès pour l'ouverture de la base :

Le fichier “usr/local/pgsql/data/pg_hba.conf” contient la configuration d'accès des utilisateurs de PostgreSQL.

L'autre fichier de configuration pour l'ouverture de la base vers l'extérieur est “postgresql.conf”.

Par défaut, le fichier “ postgresql.conf ” n'écoute que l'adresse “localhost” dans la partie “Connection and authentification”. Il faut donc activer la ligne :

```
listen_addresses = '*' (on enlève le # devant la ligne)
```

et entrer le port de connection :

```
port = 5432
```

Modification dans « /usr/local/pgsql/data/pg_hba .conf » afin de permettre l'accès vers postgres de l'extérieur :

On change à la fin du fichier dans le paragraphe #IPv4 local connections :

Host	all	all	0.0.0.0	0.0.0.0	trust
------	-----	-----	---------	---------	-------

Pour un lancement automatique de postgresql, on crée un nouveau fichier « postgresql » dans « /etc/init.d/ » par l'intermédiaire de la commande :

```
postgres@sigmas2: mkdir /etc/init.d/postgresql
```

Dans ce nouveau fichier on entre le code suivant :

```
#!/bin/sh
# postgresql      This is the init script for starting up
the PostgreSQL
#                  server

# chkconfig: - 85 15
# description: Starts and stops the PostgreSQL backend
daemon that handles all database requests.
# processname: postmaster
# pidfile: /usr/local/pgsql/data/postmaster.pid
#

# Source function library.
. /etc/rc.d/init.d/functions

# Get config.
. /etc/sysconfig/network

# Check that networking is up.
# Pretty much need it for postmaster.
[ ${NETWORKING} = "no" ] && exit 0

[ -f /usr/local/pgsql/bin/postmaster ] || exit 0

# See how we were called.
case "$1" in
start)
pid=`pidof postmaster`
if [ $pid ]
then
echo "Postmaster already running."
else
echo -n "Starting postgresql service: "
su -l postgres -c '/usr/local/pgsql/bin/pg_ctl -D
/usr/local/pgsql/data/ -l /usr/local/pgsql/data/logfile
start'
sleep 1
```

```
echo
exit
fi
;;

stop)
echo -n "Stopping postgresql service: "
killproc postmaster
sleep 2
rm -f /usr/local/pgsql/data/postmaster.pid
echo
;;

restart)
$0 stop
$0 start
;;

*)
echo "Usage: postgresql {start|stop|restart}"
exit 1
esac

exit 0
```

On rend le fichier exécutable :

```
postgres@sigmas2: chmod 777 /etc/init.d/postgresql
```

Et pour le rendre exécutable au démarrage, on tape :

```
postgres@sigmas2: chkconfig postgresql on
```

Postgres sera alors lancé à chaque démarrage de la machine.

B. PostGIS

Nous allons télécharger les fichiers du code source sur le site Internet :

<http://postgis.refractory.net/download/>

Nous choisissons la dernière version de PostGis, à savoir la version 1.1.1 (sortie le 22 janvier 2006).

Décompression des sources :

```
root@sigmas2:/usr/src# tar -zxvf postgis-1.1.1.tar.gz
```

Quand nous affichons le fichier help de la commande « configure », nous pouvons voir que PostGis nécessite la présence de bibliothèques :

```
root@sigmas2:/usr/src/postgis-1.1.1# ./configure --help
```

```
Optional Packages:
  --with-PACKAGE[=ARG]    use PACKAGE [ARG=yes]
  --without-PACKAGE       do not use PACKAGE (same as --
with-PACKAGE=no)
  --without-iconv         disable unicode support
  --with-pgsql[=ARG]      build for a specific postgresql
version
                          [ARG=path to pg_config]
  --with-geos[=ARG]       enable spatial predicates and
operators using GEOS
                          [ARG=path to geos-config]
  --with-proj[=DIR]       enable reprojection support
  --with-jts[=ARG]        use JTS for spatial predicates
and operators
                          [ARG=path to jts-config]
  --with-docdir=DIR       install the documentation in
DIR [PREFIX/share/doc]
```

Cela dépend de l'utilisation que nous faisons de PostGIS. Ainsi, en ce qui nous concerne, nous avons besoin des bibliothèques « geos » et « proj » (gestion des systèmes de projection).

1) Installation de Geos

L'installation de Geos est assez simple.

Nous nous connectons à la source de téléchargement des codes sources. Nous recherchons les bibliothèques « geos » et « libgeos » afin de choisir :

```
root@sigmas2:/usr/src/postgis-1.1.1# aptitude search geos
p      libgeos-dev                - Geometry
engine for GIS - Development files
p      libgeos-doc                -
Documentation for the GEOS GIS geometry engine li
```

```
p    libgeos2                                - Geometry
engine for Geographic Information System
```

```
root@sigmas2:/usr/src/postgis-1.1.1# aptitude search
libgeos
p    libgeos-dev                                - Geometry
engine for GIS - Development files
p    libgeos-doc                                -
Documentation for the GEOS GIS geometry engine li
p    libgeos2                                - Geometry
engine for Geographic Information System
```

Nous obtenons les mêmes fichiers lors des deux recherches.

Nous installons les fichiers « libgeos2 » et « libgeos-dev », le fichier doc ne nous intéresse pas.

Installation de la bibliothèque :

```
root@sigmas2:/usr/src/postgis-1.1.1# aptitude install
libgeos2
```

```
root@sigmas2:/usr/src/postgis-1.1.1# aptitude install
libgeos-dev
```

Vérification que la bibliothèque est bien installée :

```
root@sigmas2:/usr/src/postgis-1.1.1# geos-config
Usage: geos-config [OPTIONS]
Options:
  [--prefix]
  [--version]
  [--libs] (indique que la bibliothèque est bien
installée
  [--cflags]
  [--includes]
  [--jtsport]
```

Nous pouvons voir où est installée la bibliothèque :

```
root@sigmas2:/usr/src/postgis-1.1.1# geos-config --libs
-L/usr/lib -lgeos
```

2) Installation de Proj 4

Nous ne pouvons pas installer cette bibliothèque de la même manière car le fichier proj.dev n'est pas disponible :

```
root@sigmas2:/usr/src/postgis-1.1.1# aptitude search proj
p    autoproject                                - create a
skeleton source package for a new progra
p    egroupware-projects                        - eGroupWare
projects management application
```

```

v    libk3bproject                -
v    libk3bproject-dev            -
v    libmrproject-dev             -
v    libmrproject0                -
p    mrproject                    - project
management application
p    phpgroupware-projects        - phpGroupWare
projects management module
p    proj                        - Cartographic
projection filter and library
p    proj-ps-doc                  - PostScript
docs for cartographic projection filte
p    projectcenter                - IDE for
GNUstep Development
v    r-cran-mapproj              -
p    subproject-howto            - Debian
Subproject HOWTO

```

Il faut chercher cette bibliothèque sur Google et télécharger les sources. Nous allons sur le site Internet suivant et téléchargeons la version 4.4.9 (<http://www.remotesensing.org/proj/>)

PROJ.4 - Cartographic Projections Library

This web page relates to the PROJ.4 Cartographic Projections library originally written by Gerald Evenden then of the USGS. The primary version of this web page can be found at <http://www.remotesensing.org/proj>, and mirrored at <http://proj.maptools.org>.

Download

The current development source is available by anonymous CVS from the server ":pserver:cvsanon@cvs.maptools.org:/cvs/maptools/cvsroot". For example:

```
% cvs -d :pserver:cvsanon@cvs.maptools.org:/cvs/maptools/cvsroot login
Password:
% cvs -d :pserver:cvsanon@cvs.maptools.org:/cvs/maptools/cvsroot co proj
...
```

The following files are available from the [proj ftp directory](#) or [http mirror](#).

- [proj-4.4.9.tar.gz](#) or [proj-4.4.9.zip](#) : Current source release.
- [proj446_win32_bin.zip](#): Prebuilt Win32 executables, DLL including NAD27 grid shift files.
- [proj-datumgrid-1.3.zip](#): US, Canadian and New Zealand datum shift grids - unzip in the nad directory before configuring to add NAD27/NAD83 and NZGD49 datum conversion.
- [proj-4.4.8-1.i386.rpm](#): Binary distribution for Intel Linux (glibc 2.1). The [Mapping Hacks RPM Repository](#) may have more current PROJ.4 RPMs for Fedora Core. The [GDF Hannover RPM Repository](#) offers PROJ.4 RPMs for SuSe and Mandrake/Mandriva.
- [PROJ.4 in a Debian Package](#).
- [PROJ.4 Ported to the Delphi \(Borland C++\) environment](#).
- [PROJ.4.3.3.tar.gz](#): The last PROJ 4.3 release produced by Gerald Evenden (classic PROJ.4).
- [from kai](#): A snapshot of everything on the kai.er.usgs.gov/pub/PROJ.4 directory as of March/2000.

Documentation

Décompression du fichier :

```
root@sigmas2:/usr/src# tar -zxvf proj-4.4.9.tar.gz
```

Configuration des sources :

```
root@sigmas2:/usr/src/proj-4.4.9# ./configure
```

Installation des sources :

```
root@sigmas2:/usr/src/proj-4.4.9# make
root@sigmas2:/usr/src/proj-4.4.9# make install
```

3) Installation de PostGIS

Configuration des sources :

```
root@sigmas2:/usr/src/postgis-1.1.1# ./configure
--with-proj
--with-geos
--with-pgsql=/usr/local/pgsql/bin/pg_config
```

Rq :

« --with-pgsql = /usr/local/pgsql/bin/pg_config » : indique à PostGIS où se situe le fichier de PostGreSQL car le configurateur ne peut pas le trouver automatiquement. La configuration doit intégrer ces fichiers afin que PostGIS puisse utiliser ces bibliothèques.

Lancement de l'installation :

```
root@sigmas2:/usr/src/postgis-1.1.1# make
root@sigmas2:/usr/src/postgis-1.1.1# make install
```

III. Mapserver

A. Installation de Gdal

Téléchargement des codes sources « gdal-1.3.1.tar.gz » sur le site www.gdal.org

Décompression du fichier :

```
root@sigmas2:/usr/src# tar -zxvf gdal-1.3.1.tar.gz
```

Configuration de la bibliothèque :

```
root@sigmas2:/usr/src/gdal-1.3.1# ./configure
--with-pg = /usr/local/pgsql/bin/pg_config
--with-geos
--with-proj
```

Lancement de l'installation :

```
root@sigmas2:/usr/src/gdal-1.3.1# make
root@sigmas2:/usr/src/gdal-1.3.1# make install
```

Vérification que la bibliothèque est bien installée :

```
root@sigmas2:/usr/src/gdal-1.3.1# gdal-config
Usage: gdal-config [OPTIONS]
Options:
    [--prefix[=DIR]]
    [--libs]
    [--dep-libs]
    [--cflags]
    [--version]
    [--ogr-enabled]
    [--formats]
```

Nous pouvons voir où est installée la bibliothèque :

```
root@sigmas2:/usr/src/gdal-1.3.1# gdal-config --libs
-L/usr/local/lib -lgdal
```

Afin de mettre à jour les installations des bibliothèques, nous entrons cette commande :

```
root@sigmas2:/usr/src/gdal-1.3.1# ldconfig
```

B. Installation de Mapserver

Nous téléchargeons sur le site de Mapserver (<http://mapserver.gis.umn.edu/download/current/>) la dernière version 4.6.2 dans le dossier « sources distribution ».

Décompression du fichier :

```
root@sigmas2:/usr/src# tar -zxvf mapserver-4.6.2.tar.gz
```

Configuration des sources :

```
root@sigmas2:/usr/src/mapserver-4.6.2# ./configure
--with-proj
--with-geos
--with-ogr
--with-gdal
--with-postgis = /usr/local/pgsql/bin/pg_config
--with-httpd = /usr/local/apache2/bin/httpd
--with-php = /usr/src/php-5.1.2
--with-wfs
--with-wcs
```

Rq :

« --with-php = /usr/src/php-5.1.2 » : lance php-mapscript

« --with-wfs » et « --with-wcs » : configure Mapserver en tant que serveur. Cela pourra servir, plus tard, pour afficher des données au format image. Si nous passons par PostGIS, nous avons des données vectorielles.

Configuration des sources :

```
root@sigmas2:/usr/src/mapserver-4.6.2# make
```

Installation des sources :

Attention, nous ne faisons pas de « make install » pour Mapserver.

Nous devons copier tous les exécutables en cgi dans le répertoire « /usr/local/apache2/cgi-bin/ ».

```
root@sigmas2:/usr/src/mapserver-4.6.2/mapscript/php3
cp mapserv /usr/local/apache2/cgi-bin/
cp legend /usr/local/apache2/cgi-bin/
cp scalebar /usr/local/apache2/cgi-bin/
cp shp2img /usr/local/apache2/cgi-bin/
cp shp2pdf /usr/local/apache2/cgi-bin/
cp shptree /usr/local/apache2/cgi-bin/
cp shptreest /usr/local/apache2/cgi-bin/
cp shptreevis /usr/local/apache2/cgi-bin/
cp sortshp /usr/local/apache2/cgi-bin/
cp tile4ms /usr/local/apache2/cgi-bin/
```

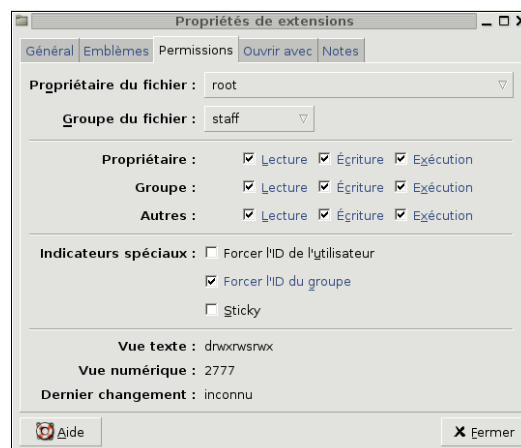
Configuration de php-mapscript :

« php-mapscript.so » est une bibliothèque qui contient les fonctions mapscripts.

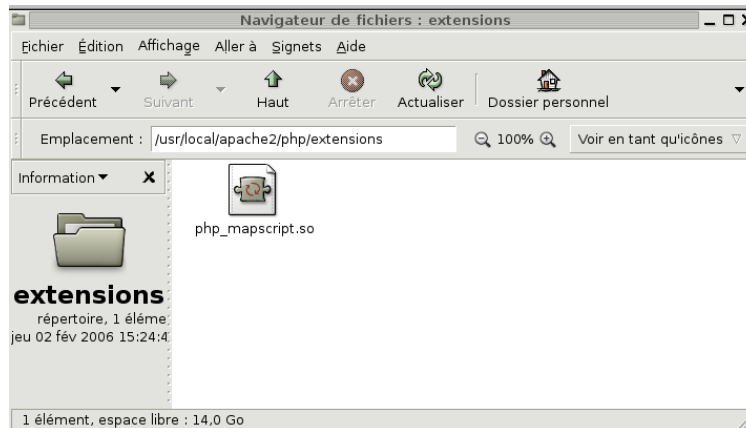
Nous créons un nouveau dossier « extensions » dans le répertoire « /usr/local/apache2/php/ »

```
4,0K drwxr-sr-x   8 root staff 4,0K 2006-02-02 15:29 .
4,0K drwxr-sr-x  16 root staff 4,0K 2006-02-02 14:24 ..
4,0K drwxr-sr-x   2 root staff 4,0K 2006-02-02 14:24 bin
4,0K drwxr-sr-x   2 root staff 4,0K 2006-02-02 14:24 etc
4,0K drwxrwsrwx   2 root staff 4,0K 2006-02-02 15:24
extensions
4,0K drwxr-sr-x   3 root staff 4,0K 2006-02-02 14:24
include
4,0K drwxr-sr-x   3 root staff 4,0K 2006-02-02 14:24 lib
4,0K drwxr-sr-x   3 root staff 4,0K 2006-02-02 14:24 man
48K -rw-r--r--   1 root staff 45K 2006-02-02 15:29
php.ini
```

Nous modifions les propriétés du dossier car il a été créé sous la session « root ».



Nous devons copier le fichier « PHP_mapscript.so » dans ce répertoire nouvellement créé.



Modification du fichier « php.ini »

Ce fichier permet de dire à PHP qui doit charger la nouvelle extension PHP-MapScript.

o première modification :

Pour l'instant, toutes les erreurs sont reportées dans le fichier log mais elles ne sont pas affichées. Si l'utilisateur ne consulte pas ce fichier, il ne connaît pas les erreurs qui ont été rencontrées.

Dans la partie du fichier « Error handling and logging », nous modifions la ligne :
« display_errors = Off » en

```
display_errors = On
```

o deuxième modification :

Nous devons indiquer à PHP le répertoire des extensions.

Dans la partie du fichier « Paths and Directories », nous rentrons la ligne suivante :

```
extension_dir = "/usr/local/apache2/php/extensions"
```

Ensuite, nous relançons Apache afin que les modifications soient prises en compte.

Nous lançons ensuite la page de Mapserver dans le navigateur Internet avec
« http://localhost/ ».

Nous obtenons un message d'erreur au sujet de la bibliothèque « libproj ». Le gestionnaire de librairie d'Apache ne trouve pas cette bibliothèque.

Dans le fichier « /usr/local/apache2/httpd.conf », il faut rajouter la ligne suivante à la fin du fichier :

```
SetEnv LD_LIBRARY_PATH /usr/lib:/usr/local/lib:  
/usr/local/pgsql/lib
```

On rajoute également dans le fichier « /etc/ld.so.conf » les lignes suivantes :

```
/usr/lib  
/usr/local/lib  
/usr/local/pgsql/lib
```

On crée un fichier temporaire qui relate de l'ensemble des chemins d'accès des librairies disponibles, pour se faire la ligne de commande suivante doit être tapé :

```
Export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:/usr/local/lib:  
/usr/localpgsql/lib
```

Afin de mettre à jour les installations des bibliothèques, nous entrons cette commande :

```
Ldconfig
```

Test pour vérifier que PHP et MapScript fonctionnent bien : nous modifions le fichier « test.php » situé dans le dossier "/var/www" qui contient les informations suivantes :

```
<?php  
  
dl("php_mapscript.so");  
  
php.info();  
  
?>
```

Quand on lance l'ouverture de la page sous MapServer, la bibliothèque MapScript est indiquée.

MapScript

MapServer Version	MapServer version 4.4.2 OUTPUT=GIF OUTPUT=PNG OUTPUT=JPEG OUTPUT=WBMP OUTPUT=PDF OUTPUT=SWF SUPPORTS=PROJ SUPPORTS=FREETYPE SUPPORTS=WMS_SERVER SUPPORTS=WMS_CLIENT SUPPORTS=WFS_SERVER SUPPORTS=WFS_CLIENT SUPPORTS=WCS_SERVER INPUT=EPPL7 INPUT=POSTGIS INPUT=OGR INPUT=GDAL INPUT=SHAPEFILE
PHP MapScript Version	(\$Revision: 1.220.2.3 \$ \$Date: 2005/02/28 14:27:19 \$)

Quelques commandes utiles

`Cd` : permet de se déplacer

{
 `..` : permet de remonter d'un répertoire
 `/` : [nom de fichier] permet de descendre dans un répertoire (les crochets ne sont pas à mettre)
 `cd ..` : remonte d'un répertoire.
 `cd usr/src/` : permet de se diriger dans le répertoire « `/usr/src/` »

`chown` : change le nom du propriétaire

`dir src` : permet de montrer les fichiers présents à l'intérieur du dossier « `src` »

`mkdir src` : permet de créer le dossier « `src` »

`Chmod 777 « nom de fichier » -R` : veut dire changement de mode en lecture, écriture et exécuter permis, au nom de fichier suivant. Le `-R` indique que l'ensemble des fichiers présents à l'intérieur du dossier, seront également modifiés.

ANNEXE 2

GUIDE EXPLICATIF DE CRÉATION DE LA BASE DE DONNÉES

Sommaire

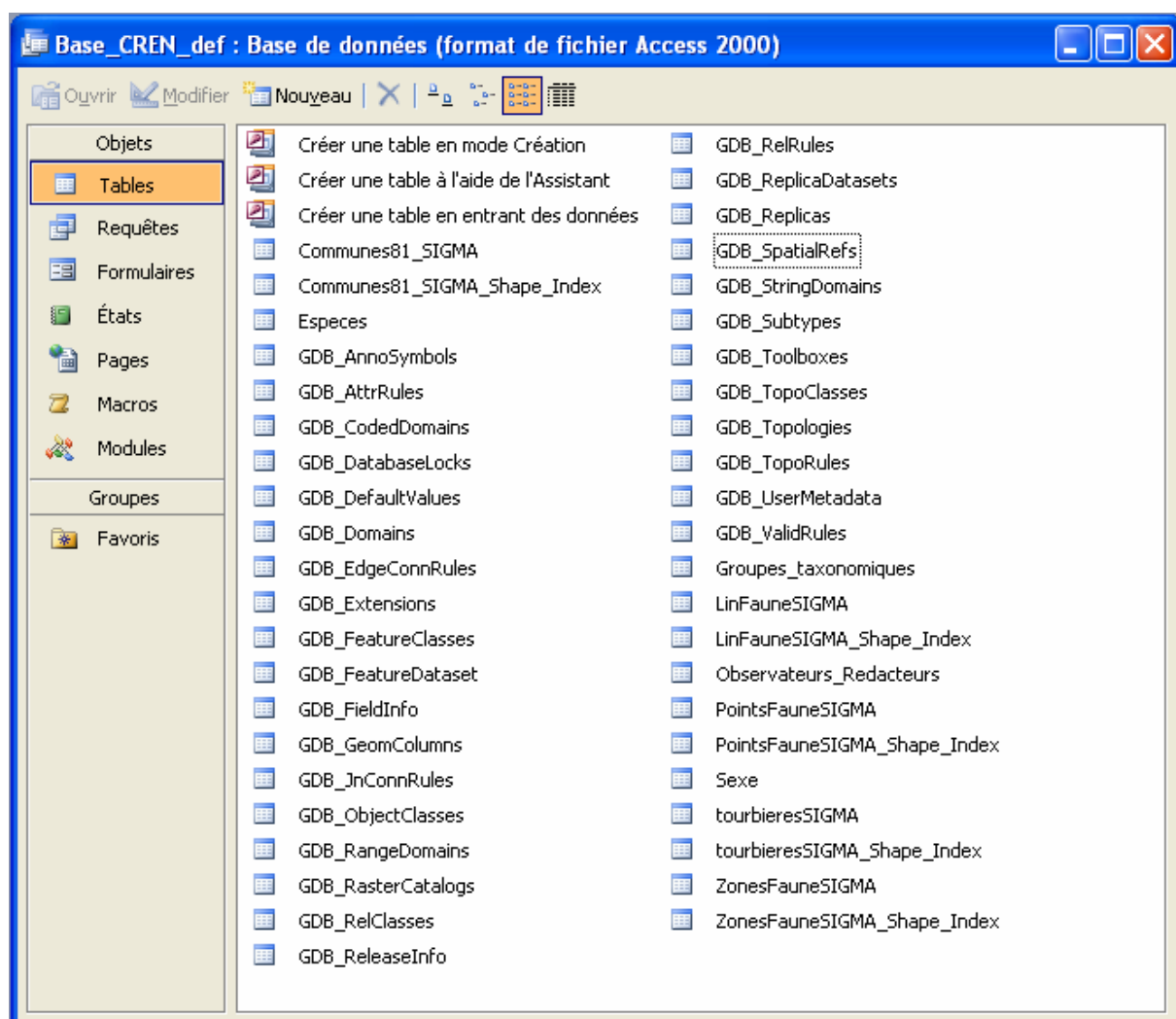
I.	STRUCTURATION DES DONNEES SUR ACCESS.....	52
A.	INTEGRATION DES SHAPES DANS LA GEODATABASE.....	52
B.	MODELE PHYSIQUE DE DONNEES	53
II.	INSTALLATION DU LOGICIEL D'ADMINISTRATION DE LA BASE DE DONNEES	54
III.	CONFIGURATION DE POSTGRESQL/POSTGIS.....	58
IV.	INTEGRATION DES DONNEES SOUS POSTGIS	61
A.	AJOUT DU DRIVER ODBC :.....	61
B.	CONFIGURATION DE LA CONNEXION	62
1)	<i>Etape 1 : transfert de la base sur le poste client.....</i>	<i>62</i>
a.	Access vers PostGres sur le poste client :	62
b.	ArcView vers PostGIS sur le poste client :	65
2)	<i>Etape 2 : transfert de la base sur le serveur</i>	<i>71</i>
a.	PostGreSQL du poste local vers le serveur :	71
b.	PostGIS du poste local vers le serveur :	73
V.	PASSERELLE D'EXPORTATION DES DONNEES	75
A.	EXPORT DES DONNEES ATTRIBUTAIRES	75
1)	<i>Le driver ODBC.....</i>	<i>75</i>
2)	<i>Configuration de la connexion.....</i>	<i>76</i>
B.	EXPORT DES DONNEES GRAPHIQUES	81

I. Structuration des données sur Access

A. *Intégration des shapes dans la Géodatabase*

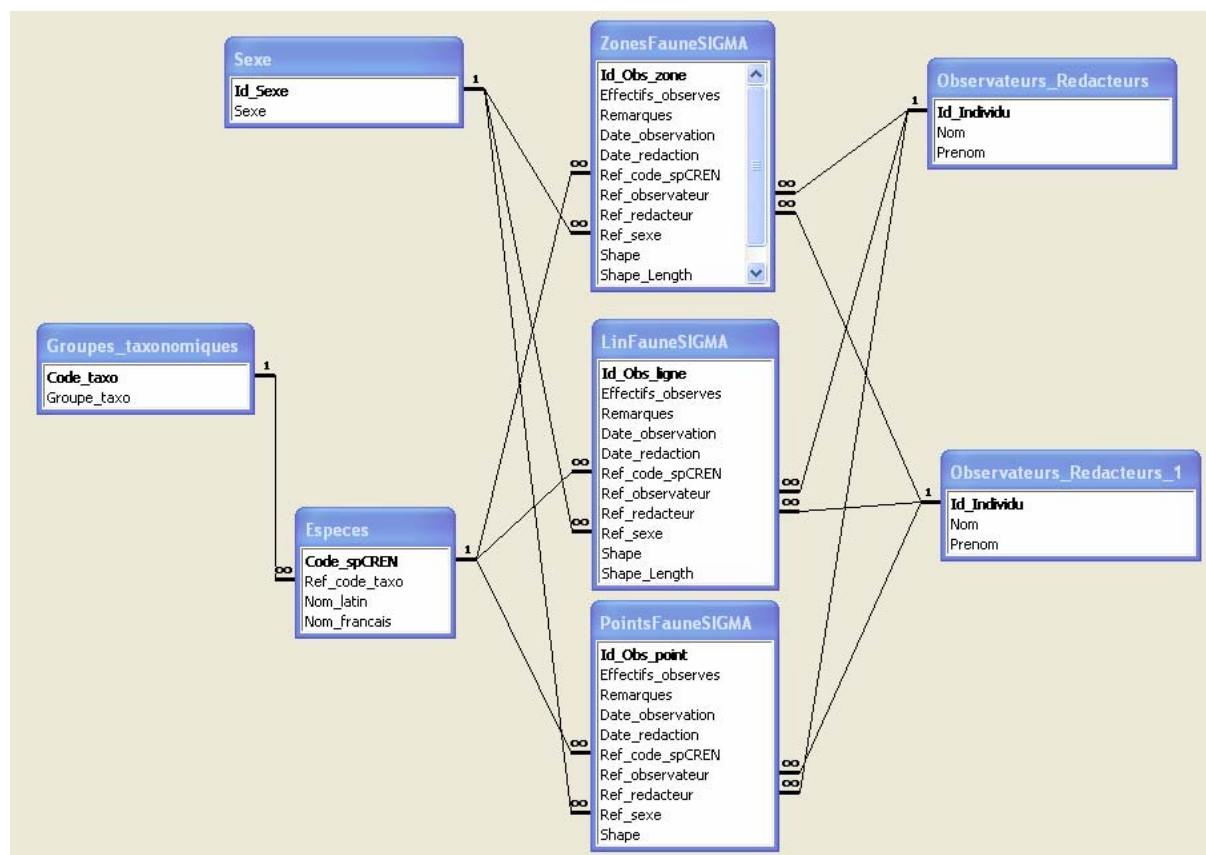
Afin de simplifier l'intégration des données dans la base PostGreSQL/PostGIS sur le serveur distant, nous les avons tout d'abord structurées sous Access à partir de notre Modèle Logique de Données.

Nous avons des données géographiques sous ArcGIS (au format .shp) que nous avons importées dans notre base Access sous forme de Géodatabase. Nous avons des tables stockant la géométrie.



B. Modèle Physique de Données

Voici les tables avec leurs relations :



La table « observation » est composée de 3 tables : une table par type de géométrie (points, lignes, polygones). Chacune de ces tables est structurée de la même manière.

Deux solutions s'offraient à nous pour traduire la table « observation » :

- Une seule table regroupant les données attributaires. Un champ supplémentaire permet de faire le lien avec trois tables contenant uniquement la géométrie.
- Trois tables stockant à la fois les données attributaires et la géométrie.

Il nous a semblé plus simple pour les chargés d'études de mettre en œuvre la deuxième solution. La saisie des données géographiques ainsi que des données attributaires se font toutes dans la même table, ce qui est plus aisé pour des personnes non-spécialistes.

La table « observateur / rédacteur » stocke les informations concernant le personnel du CREN. Ils peuvent être observateur et/ou rédacteur (ce dernier étant celui qui va saisir les informations dans la base).

Dans notre MPD, la même table apparaît deux fois :

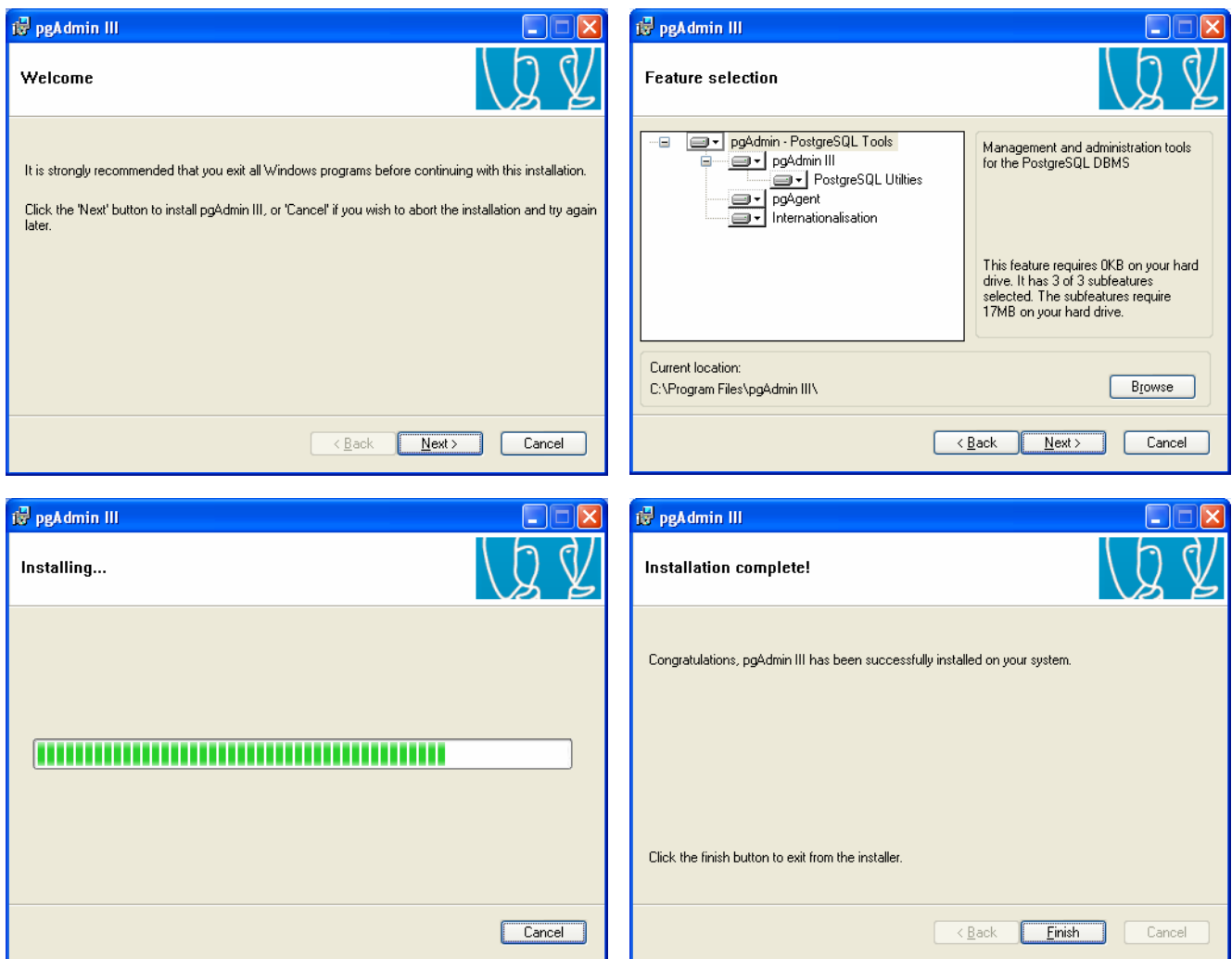
- la première permet de relier l'individu qui a fait l'observation à l'une des tables d'observation (« Ref_observateur » des tables Faunes)
- la seconde permet de relier l'individu qui saisit les données à la table d'observation correspondante (« Ref_redacteur » des tables Faunes)

Afin de gérer la base de données PostgreSQL/PostGIS, nous avons choisi d'utiliser un logiciel libre. Il existe deux possibilités pour la gestion de la base :

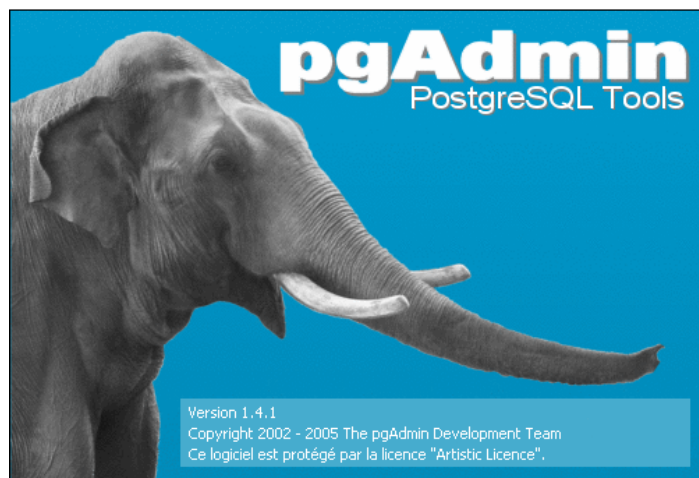
- L'installation de phpPgAdmin étant délicate et moins stable sur le serveur Linux, nous avons opté pour l'utilisation de PgAdmin sur le poste client.

suivante : <http://www.postgresql.org/ftp/pgadmin3/release/v1.4.1/win32/>

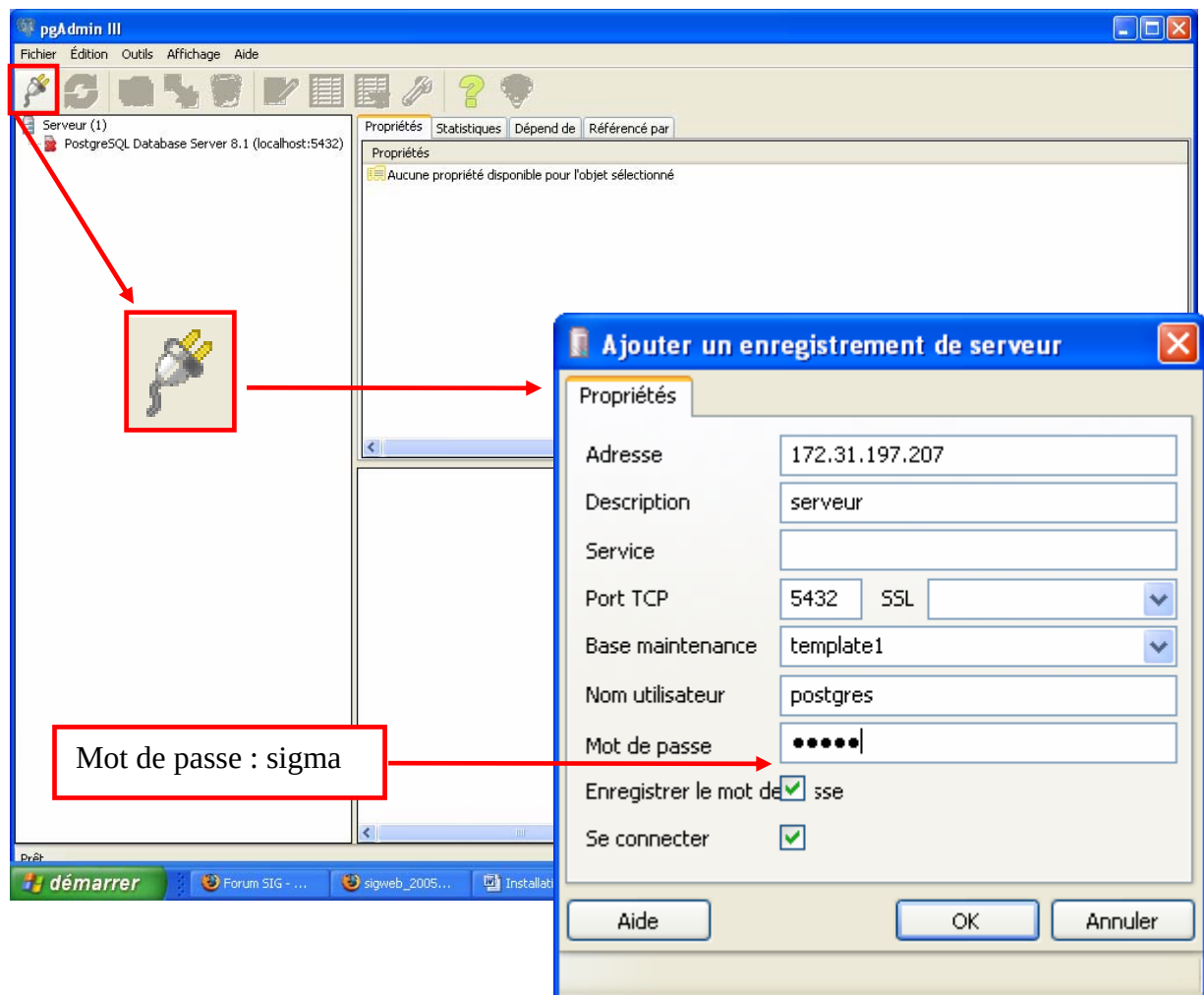
Après avoir décompressé le dossier, nous lançons l'installation de logiciels.



Nous lançons l'application.



Configuration de la connexion à notre serveur :



La connexion au serveur ne passe pas car le poste client n'est pas autorisé à se connecter à la base. Il faut donc modifier le fichier de configuration de la connexion à la base de données sur le serveur. Ce fichier « pg_hba.conf » se trouve dans le dossier « /usr/local/pgsql/data/ ».

A la fin du fichier, il faut donner l'autorisation à tous les postes de se connecter au serveur. Pour cela, nous devons modifier la ligne de la manière suivante :

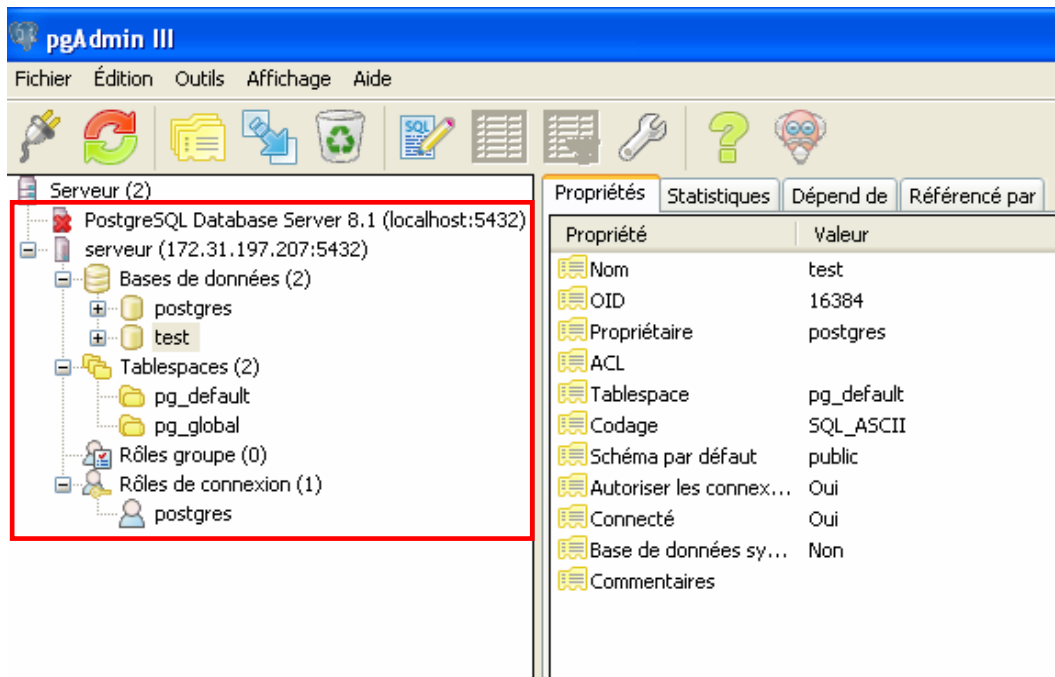
```
host all 0.0.0.0 0.0.0.0 trust
```

Il faut ensuite arrêter et redémarrez PostGreSQL :

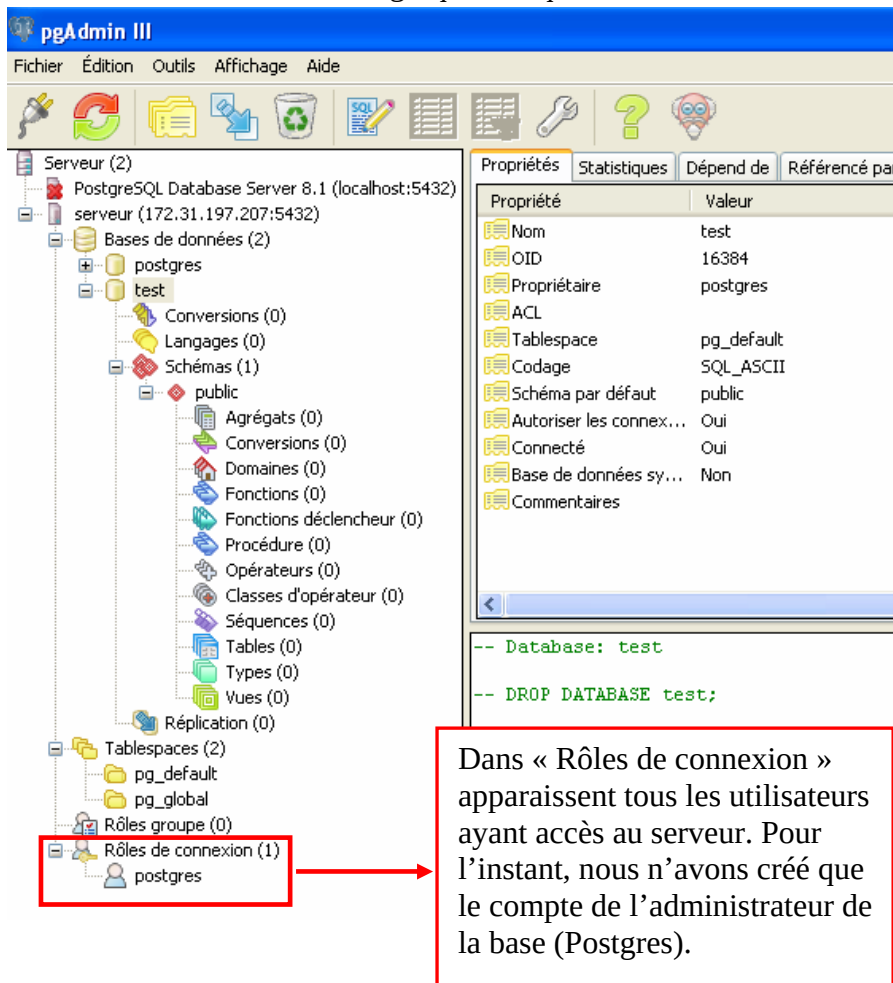
```
postgres@sigmas2: /usr/local/pgsql/bin/pg_ctl stop -D /usr/local/pgsql/data
```

```
postgres@sigmas2: /usr/local/pgsql/bin/pg_ctl start -D /usr/local/pgsql/data
```

A l'issue de cette modification, une nouvelle connexion est ajoutée dans l'explorateur (situé à gauche de la fenêtre). Elle reprend les éléments créés lors de l'installation de PostgreSQL sur le serveur. Ainsi, nous retrouvons la base de données « test ».

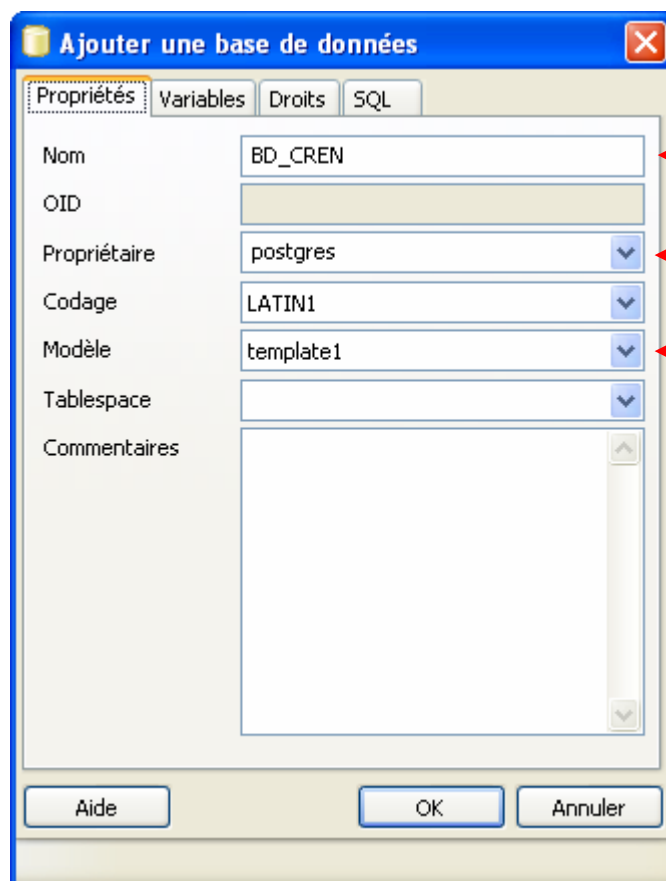
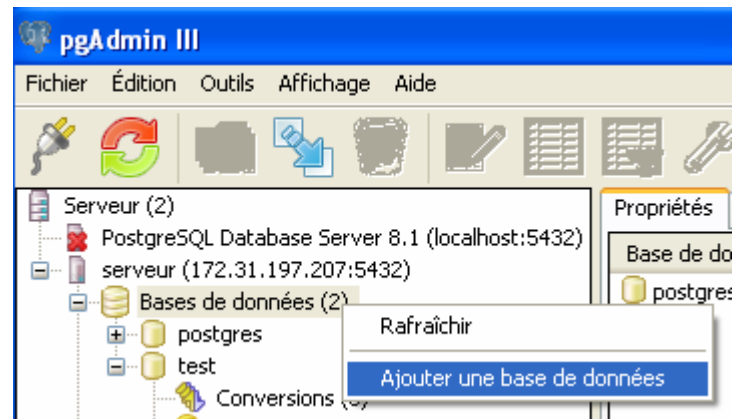


L'architecture de la base est proposée par défaut. Le schéma public contient l'ensemble des tables. Pour l'instant, il ne s'agit que du squelette des tables.



III. Configuration de PostgreSQL/PostGIS

Maintenant que le logiciel est installé et configuré, nous pouvons lancer la création de la base de données.



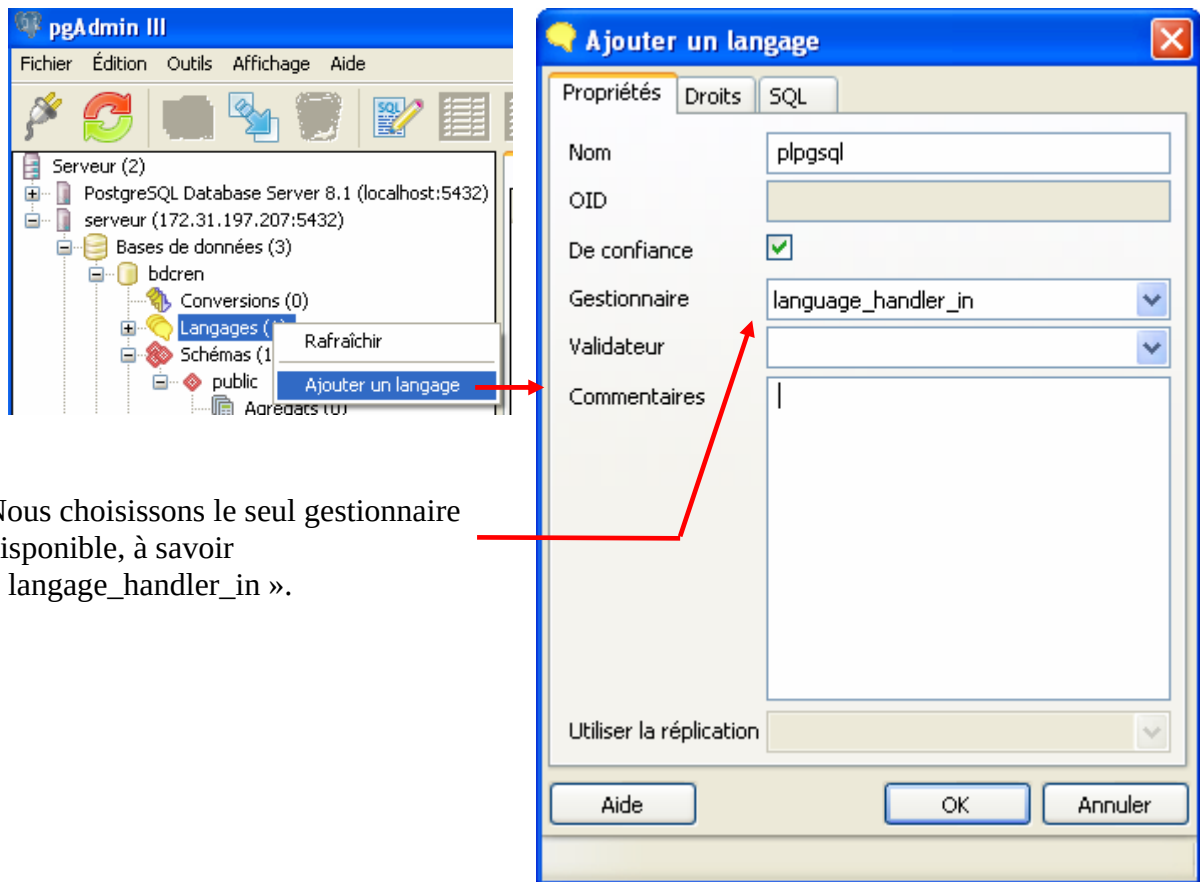
Nom de la base de données

Nom de l'administrateur de la base

Template 1 : modèle qui contient les extensions PostGIS

Nous avons utilisé le template1 mais les fonctionnalités de PostGIS ne sont pas encore activées sur le serveur Linux.

Tout d'abord, nous ajoutons le langage pl/pgsql (langage propre à PostgreSQL) à notre base de données.



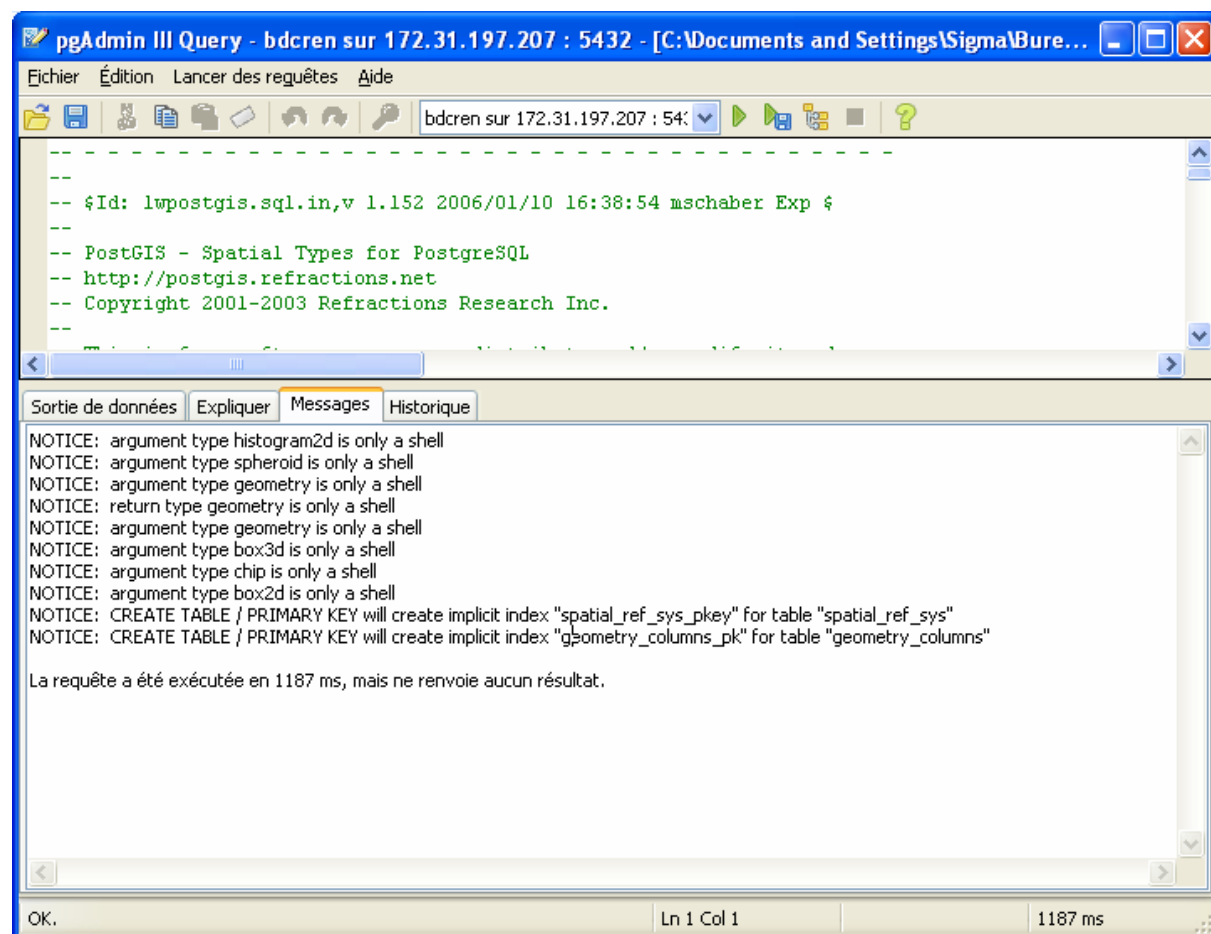
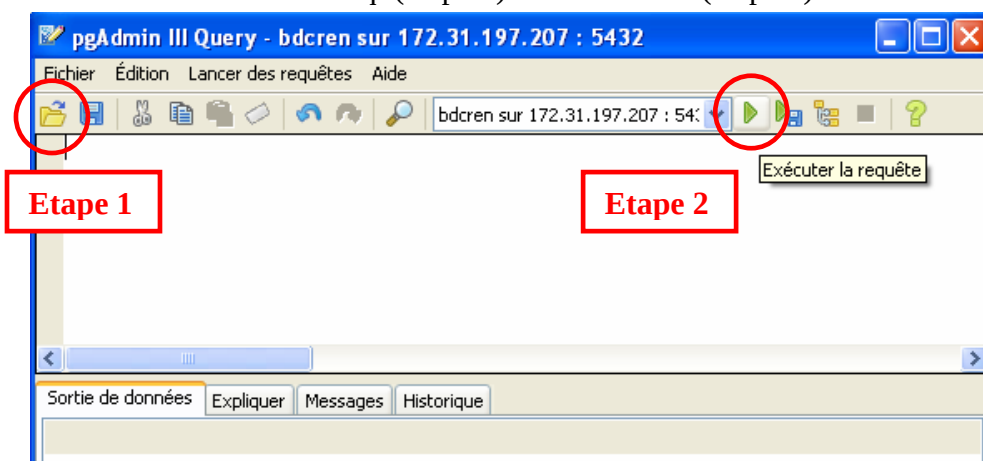
Nous choisissons le seul gestionnaire disponible, à savoir « language_handler_in ».

Ensuite, nous activons les fonctions de PostGIS avec deux fichiers « lwpostgis.sql » et « spatial_ref_sys.sql ». Nous les recherchons sur le serveur et les lançons sur le poste client dans pgAdmin.

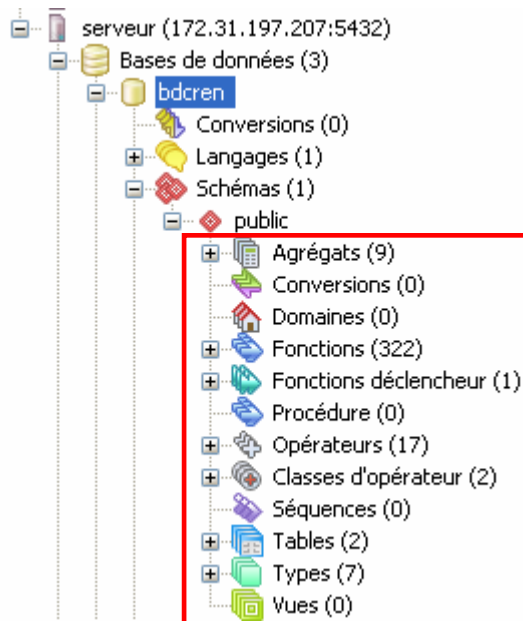
Pour cela, nous cliquons sur l'icône « Exécuter vos propres requêtes SQL » dans la barre d'outils du menu principal



Nous ouvrons le fichier .sql (étape 1) et l'exécutons (étape 2).



Quand nous regardons l'explorateur de fichier, nous observons l'ajout des fonctionnalités de PostGIS.



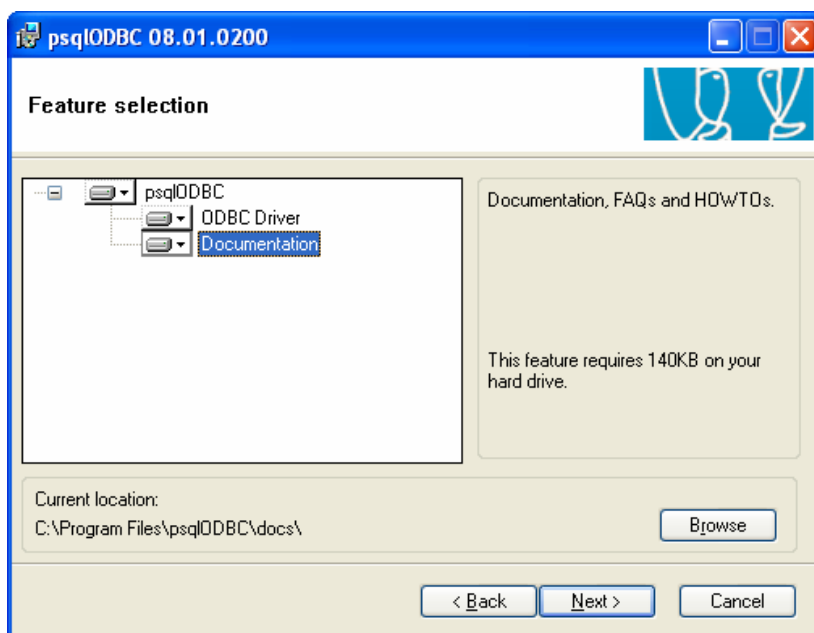
IV. Intégration des données sous PostGIS

A. *Ajout du driver ODBC :*

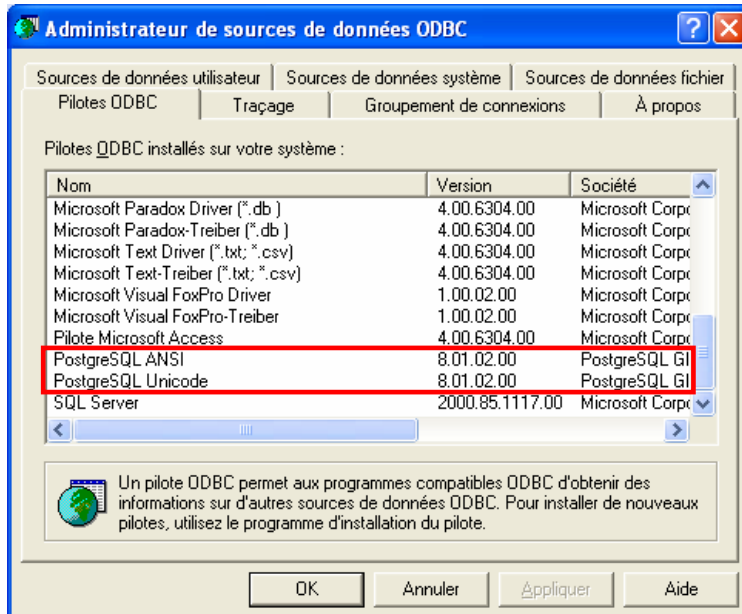
Nous allons configurer la connexion entre notre base de données Access et notre base PostGreSQL/PostGIS grâce à un lien ODBC.

Nous téléchargeons le driver odbc pour PostGreSQL sur le site suivant :

<http://www.postgresql.org/ftp/odbc/versions/msi/>



Après avoir installé le pilote, nous pouvons voir qu'il a été rajouté dans la fenêtre « Administration de sources de données ODBC » accessible dans le « Panneau de configuration / Outils d'administration ».

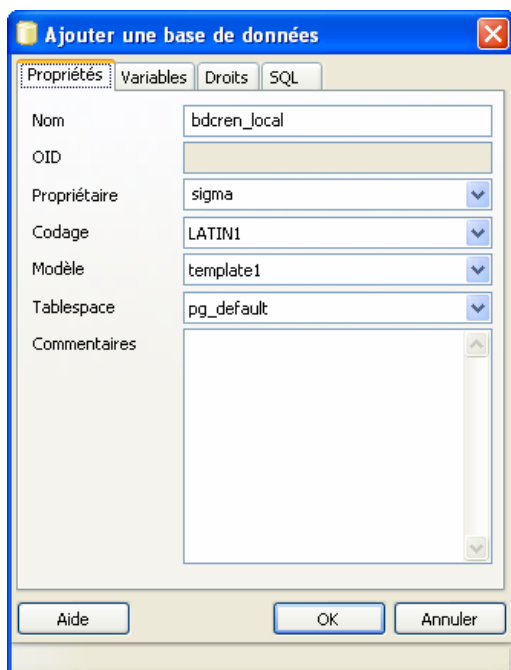


B. Configuration de la connexion

Nous allons transférer la base de données du SGBD Microsoft Access à PostGreSQL en local sur le poste client. Puis, nous l'exporterons sur PostGreSQL situé sur le serveur. Cela permettra de limiter les erreurs de transfert et, le cas échéant, d'isoler les sources d'erreur.

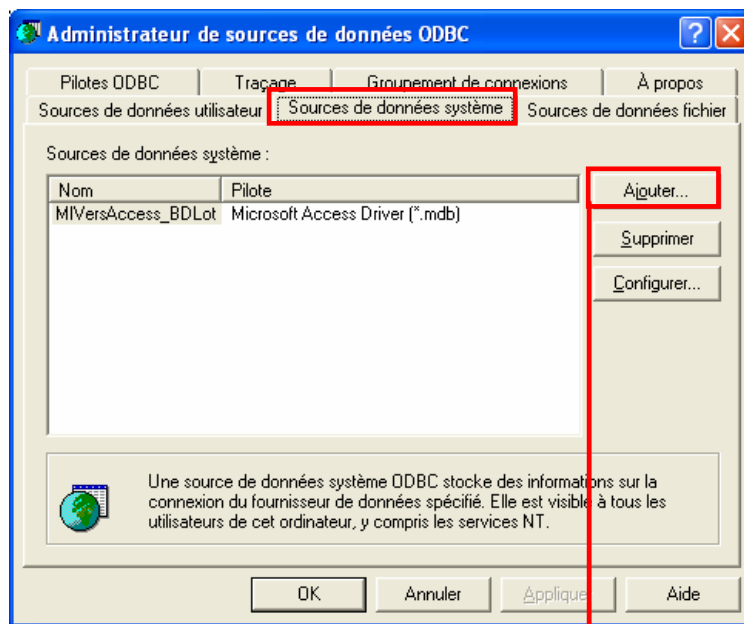
1) Etape 1 : transfert de la base sur le poste client

a. Access vers PostGres sur le poste client :

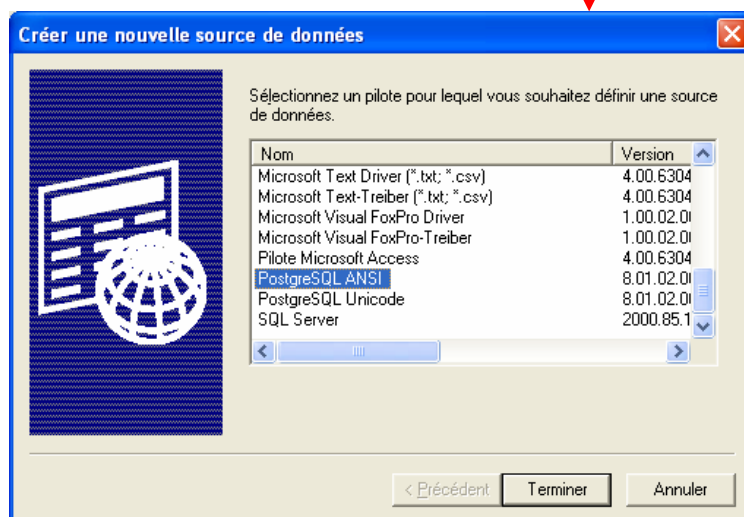


Tout d'abord, nous allons créer notre base de données en local.

Configuration du DSN Access vers Postgres :

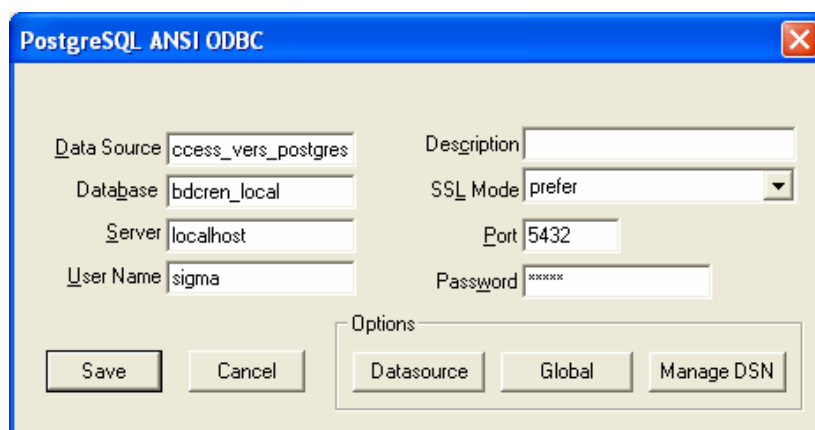


Nous choisissons de créer une connexion dans l'onglet « Sources de données système ». Ce type de configuration permet d'accéder à ce DSN pour tous les utilisateurs de la machine.



Nous sélectionnons ici le driver « PostgreSQL ANSI ». C'est le pilote qui gère le mieux les caractères accentués sous Windows.

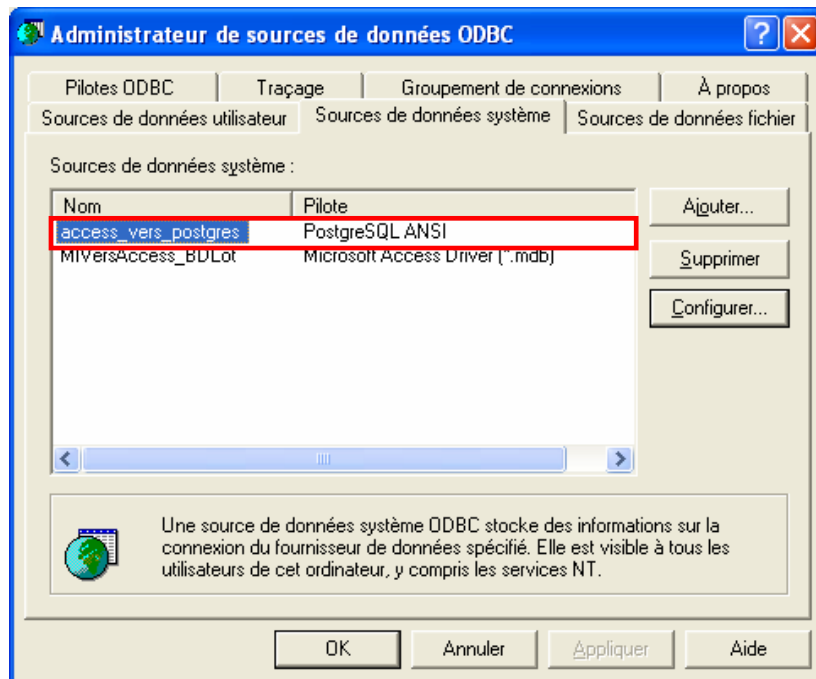
Paramètres de connexion avec la base de données locale sur PostgreSQL : « bdcresn_local »



Data Source : nom de la connexion.

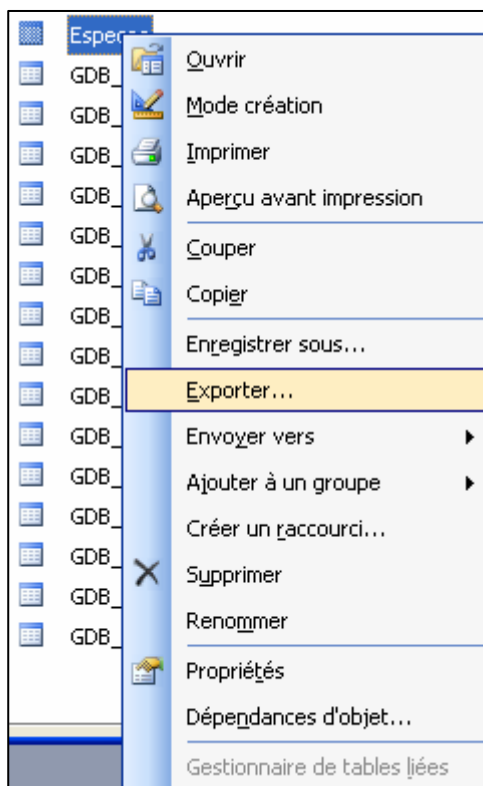
Database : nom de notre base locale

Server : en local (« localhost »)

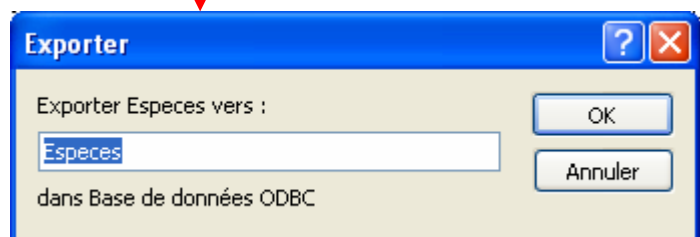
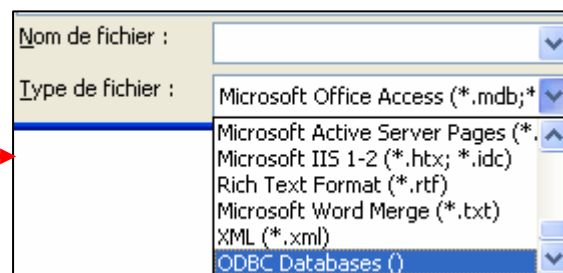


Notre nouvelle connexion s'est ajoutée dans l'onglet « Source de données système ».

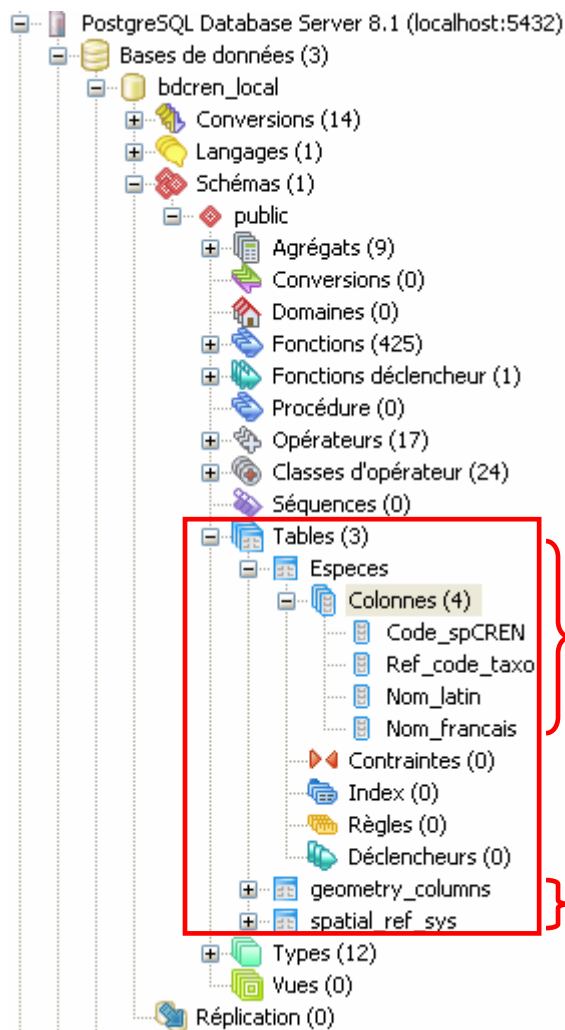
Dans Access, nous exportons les tables attributaires une par une en faisant « clic droit / exporter... » sur la table.



Nous choisissons ici notre type de fichier, à savoir « ODBC Databases () »



La table « especes » a été créée dans la base « bdcren_local » sur le poste client.



Nous retrouvons bien la table « especes » composée de 4 champs,

ainsi que les deux tables (créées automatiquement avec le choix du « template 1 » définissant la structure par défaut utilisant PostGIS.

b. ArcView vers PostGIS sur le poste client :

Nous devons tout d'abord remplir les champs des tables « Faunes » qui seront définis comme clé étrangère. En effet, ils sont actuellement remplis par des « zéros ». Nous allons mettre de fausses valeurs afin de pouvoir intégrer ces tables.

Procédé : l'objectif est de convertir le fichier Shape en un script SQL qui pourra être exécuté par PgAdmin. La conversion du fichier Shape en script SQL se fait par l'intermédiaire d'un utilitaire installé par défaut avec PostGIS : **shp2pgsql.exe**

L'importation d'un fichier Shape sous PostGIS se fait donc en trois étapes :

- 1) La création d'un fichier SQL qui comportera la requête de création de la table PostGIS correspondant à notre fichier SHAPE
- 2) L'exécution de cette requête sous PgAdmin qui va créer physiquement la table correspondante.
- 3) Enfin il faut créer un index sur le champ comportant la géométrie afin de faciliter l'exécution des requêtes spatiales.

Création du script SQL :

Cette étape se déroule sous l'invite de commande de Windows (Menu « Démarrer/programmes/Accessoires/Invite de commandes »).

Il faut se placer dans le dossier qui contient les fichiers shape à intégrer. La structure de la commande à taper pour exécuter le programme est de type :

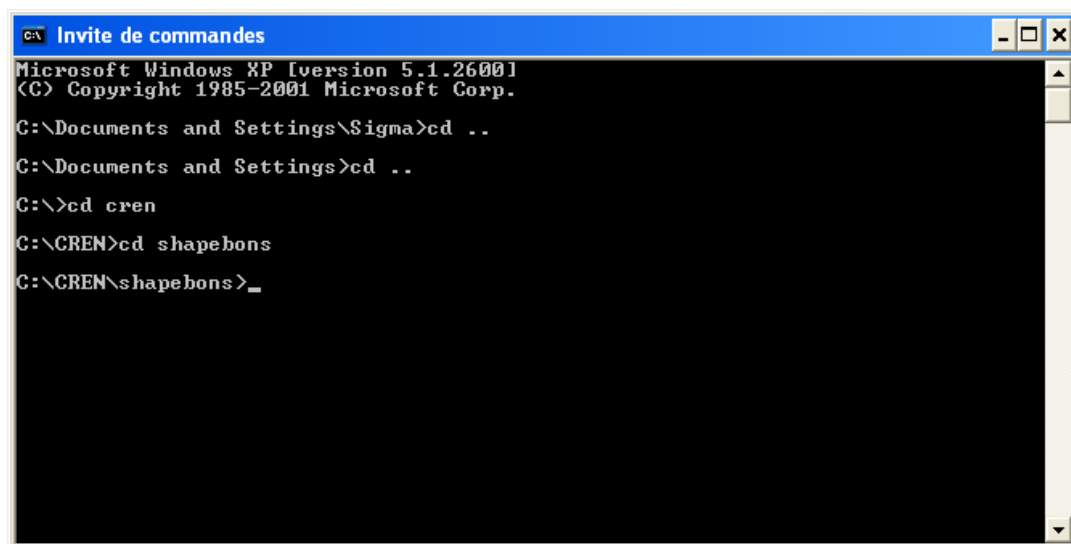
Outil_de_conversion -s **codeEPSG** **shapefile** **nomtable** > **fichier_à_créer.sql**

Avec :

- **Outil_de_conversion** : L'utilitaire de conversion est cité plus haut (**shp2pgsql**). Il faut signifier l'emplacement de l'utilitaire **entre guillemet** qui est par défaut « C:/Program Files/PostgreSQL/8.1/bin/shp2pgsql.exe »
- **-s codeEPSG** : C'est le code de la projection du fichier Shape. Le code du Lambert 2 carto est le 27592 (tous les codes EPSG sont disponibles sur www.inovagis.org)
- **Shapefile** : nom du fichier Shape à importer
- **nomtable** : nom de la nouvelle table créée
- **fichier_à_créer.sql** : nom du fichier SQL créé.

Nous allons montrer en détail la création du script SQL pour le fichier Shape « ZonesfauneSIGMA » situé dans « c:/CREN/shapebons »

Placez vous dans le répertoire voulu :

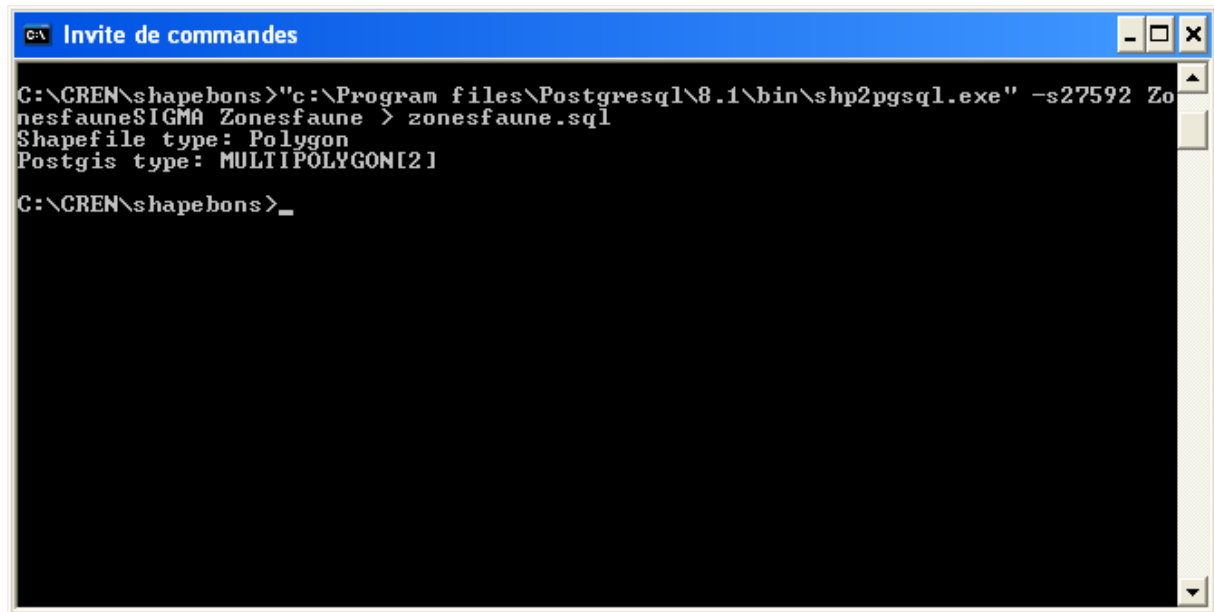


```
C:\ Invite de commandes
Microsoft Windows XP [version 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.

C:\Documents and Settings\Sigma>cd ..
C:\Documents and Settings>cd ..
C:\>cd cren
C:\CREN>cd shapebons
C:\CREN\shapebons>_
```

Une fois placé dans le bon dossier il faut traduire la formule plus haut en l'adaptant à notre cas de figure :

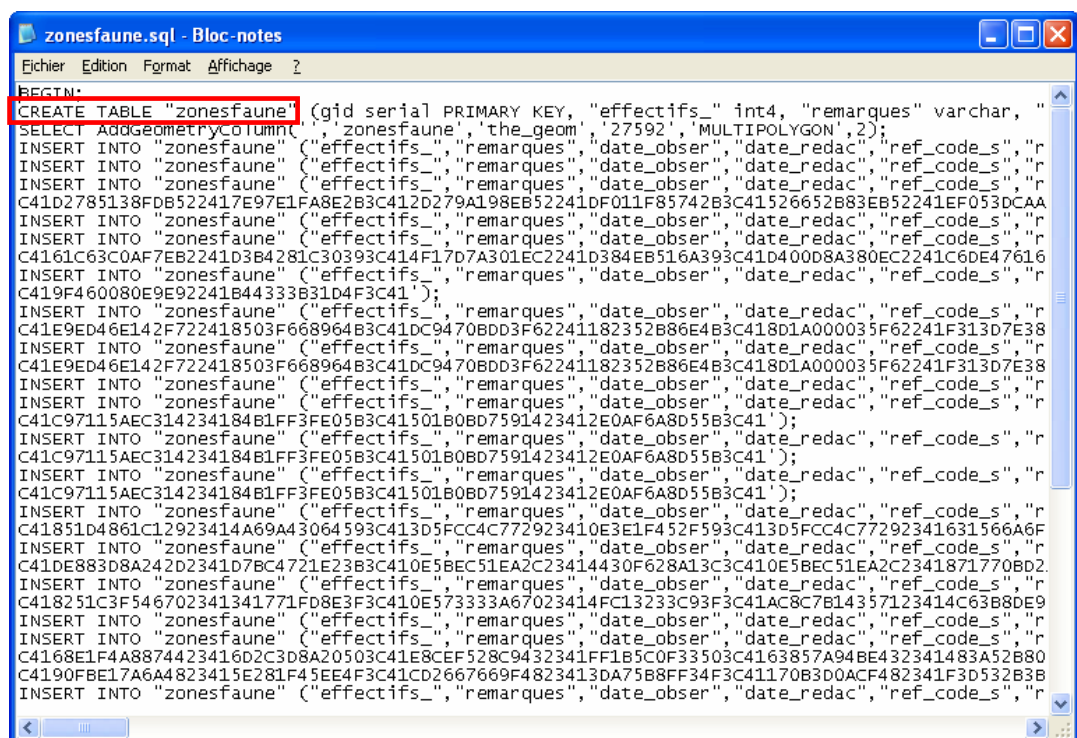
« C:/Program Files/PostGreSQL/8.1/bin/shp2pgsql.exe » -s 27592 ZonesfauneSIGMA
Zonesfaune > Zonesfaune.sql



```
C:\CREN\shapebons>"c:\Program files\Postgresql\8.1\bin\shp2pgsql.exe" -s27592 ZonesfauneSIGMA Zonesfaune > zonesfaune.sql
Shapefile type: Polygon
Postgis type: MULTIPOLYGON[2]
C:\CREN\shapebons>
```

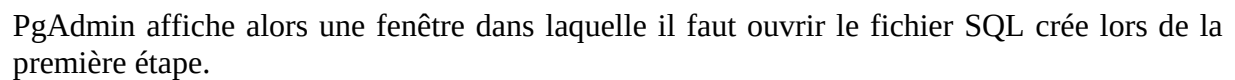
On constate que le type du fichier Shape a bien été reconnu en tant que « polygone »

On peut vérifier ensuite que le fichier a bien été créé dans le dossier et qu'il correspond bien à une requête de création de table.

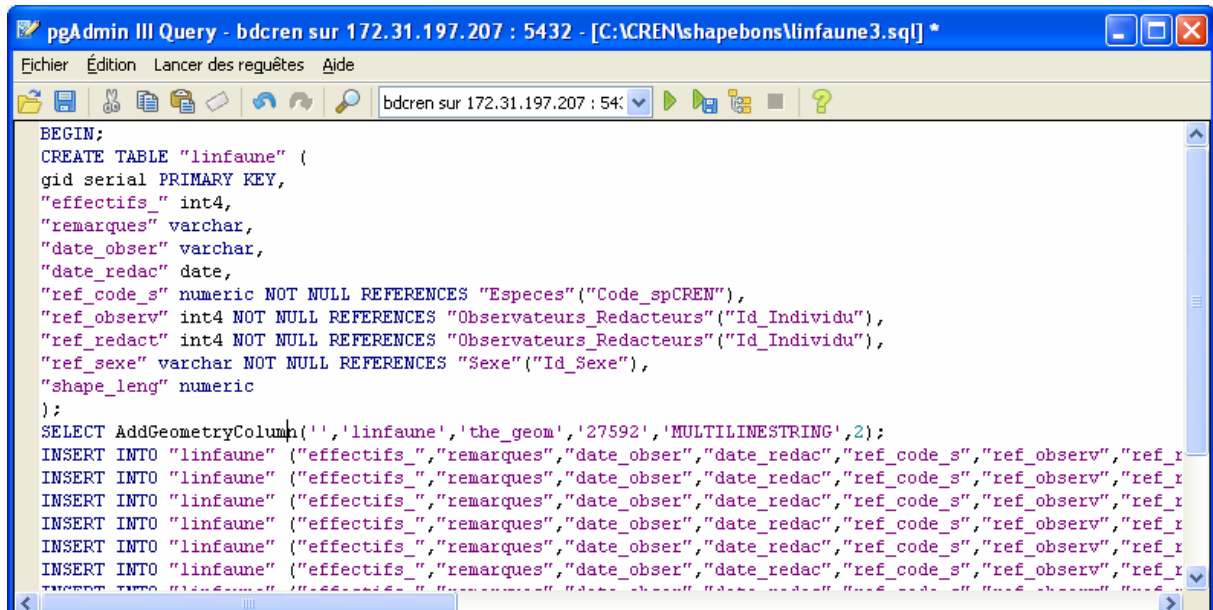


```
zonesfaune.sql - Bloc-notes
Fichier Edition Format Affichage ?
BEGIN;
CREATE TABLE "zonesfaune" ("gid serial PRIMARY KEY, "effectifs_" int4, "remarques" varchar, "
SELECT AddGeometryColumn("zonesfaune","the_geom","27592","MULTIPOLYGON",2);
INSERT INTO "zonesfaune" ("effectifs_", "remarques", "date_obs", "date_redac", "ref_code_s", "r
INSERT INTO "zonesfaune" ("effectifs_", "remarques", "date_obs", "date_redac", "ref_code_s", "r
INSERT INTO "zonesfaune" ("effectifs_", "remarques", "date_obs", "date_redac", "ref_code_s", "r
C41D2785138FDB522417E97E1FA8E2B3C412D279A198EB52241DF011F85742B3C41526652B83EB52241EF053DCAA
INSERT INTO "zonesfaune" ("effectifs_", "remarques", "date_obs", "date_redac", "ref_code_s", "r
INSERT INTO "zonesfaune" ("effectifs_", "remarques", "date_obs", "date_redac", "ref_code_s", "r
C4161C63C0AF7EB2241D3B4281C30393C414F17D7A301EC2241D384EB516A393C41D400D8A380EC2241C6DE47616
INSERT INTO "zonesfaune" ("effectifs_", "remarques", "date_obs", "date_redac", "ref_code_s", "r
C419F460080E9E2241B44333B31D4F3C41');
INSERT INTO "zonesfaune" ("effectifs_", "remarques", "date_obs", "date_redac", "ref_code_s", "r
C41E9ED46E142F722418503F668964B3C41DC9470BDD3F62241182352B86E4B3C418D1A000035F62241F313D7E38
INSERT INTO "zonesfaune" ("effectifs_", "remarques", "date_obs", "date_redac", "ref_code_s", "r
C41E9ED46E142F722418503F668964B3C41DC9470BDD3F62241182352B86E4B3C418D1A000035F62241F313D7E38
INSERT INTO "zonesfaune" ("effectifs_", "remarques", "date_obs", "date_redac", "ref_code_s", "r
INSERT INTO "zonesfaune" ("effectifs_", "remarques", "date_obs", "date_redac", "ref_code_s", "r
C41C97115AEC314234184B1FF3FE05B3C41501B0BD7591423412E0AF6A8D55B3C41');
INSERT INTO "zonesfaune" ("effectifs_", "remarques", "date_obs", "date_redac", "ref_code_s", "r
C41C97115AEC314234184B1FF3FE05B3C41501B0BD7591423412E0AF6A8D55B3C41');
INSERT INTO "zonesfaune" ("effectifs_", "remarques", "date_obs", "date_redac", "ref_code_s", "r
C41C97115AEC314234184B1FF3FE05B3C41501B0BD7591423412E0AF6A8D55B3C41');
INSERT INTO "zonesfaune" ("effectifs_", "remarques", "date_obs", "date_redac", "ref_code_s", "r
C41851D4861C12923414A69A43064593C413D5FCC4C772923410E3E1F452F593C413D5FCC4C77292341631566A6F
INSERT INTO "zonesfaune" ("effectifs_", "remarques", "date_obs", "date_redac", "ref_code_s", "r
C41DE883D8A242D2341D7BC4721E23B3C410E5BEC51EA2C23414430F628A13C3C410E5BEC51EA2C2341871770BD2
INSERT INTO "zonesfaune" ("effectifs_", "remarques", "date_obs", "date_redac", "ref_code_s", "r
C418251C3F546702341341771FD8E3F3C410E57333A67023414FC13233C93F3C41AC8C7B14357123414C63B8DE9
INSERT INTO "zonesfaune" ("effectifs_", "remarques", "date_obs", "date_redac", "ref_code_s", "r
INSERT INTO "zonesfaune" ("effectifs_", "remarques", "date_obs", "date_redac", "ref_code_s", "r
C4168E1F4A8874423416D2C3D8A20503C41E8CEF528C9432341FF1B5C0F33503C4163857A94BE432341483A52B80
C4190FBE17A6A4823415E281F45EE4F3C41CD2667669F4823413DA75B8FF34F3C41170B3D0ACF482341F3D532B3B
INSERT INTO "zonesfaune" ("effectifs_", "remarques", "date_obs", "date_redac", "ref_code_s", "r
```

Ouvrez PGAdmin III et placez vous sur votre base de données. Cliquez ensuite sur l'icône SQL de la barre d'outils



C'est ici que nous définissons les clés primaires et étrangères, ainsi que les autres contraintes d'intégrités.

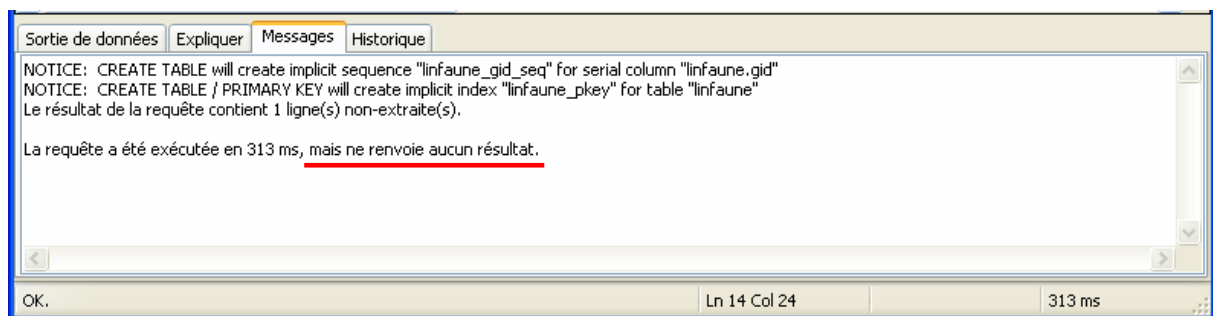


```
BEGIN;
CREATE TABLE "linfaune" (
gid serial PRIMARY KEY,
"effectifs" int4,
"remarques" varchar,
"date_obser" varchar,
"date_redac" date,
"ref_code_s" numeric NOT NULL REFERENCES "Especes"("Code_spCREN"),
"ref_observ" int4 NOT NULL REFERENCES "Observateurs_Redacteurs"("Id_Individu"),
"ref_redact" int4 NOT NULL REFERENCES "Observateurs_Redacteurs"("Id_Individu"),
"ref_sexe" varchar NOT NULL REFERENCES "Sexe"("Id_Sexe"),
"shape_leng" numeric
);
SELECT AddGeometryColumn('','linfaune','the_geom','27592','MULTILINESTRING',2);
INSERT INTO "linfaune" ("effectifs","remarques","date_obser","date_redac","ref_code_s","ref_observ","ref_r
INSERT INTO "linfaune" ("effectifs","remarques","date_obser","date_redac","ref_code_s","ref_observ","ref_r
INSERT INTO "linfaune" ("effectifs","remarques","date_obser","date_redac","ref_code_s","ref_observ","ref_r
INSERT INTO "linfaune" ("effectifs","remarques","date_obser","date_redac","ref_code_s","ref_observ","ref_r
INSERT INTO "linfaune" ("effectifs","remarques","date_obser","date_redac","ref_code_s","ref_observ","ref_r
INSERT INTO "linfaune" ("effectifs","remarques","date_obser","date_redac","ref_code_s","ref_observ","ref_r
INSERT INTO "linfaune" ("effectifs","remarques","date_obser","date_redac","ref_code_s","ref_observ","ref_r
```

Une fois la requête SQL chargée et les contraintes rajoutées, il ne reste plus qu'à l'exécuter en cliquant dans la barre d'outils sur l'icône :



Si l'exécution de la requête se déroule bien, PGAdmin affiche alors dans l'onglet « messages » le compte-rendu.



Il est normal de voir s'afficher « la requête ne renvoie aucun résultat » car c'est une simple requête de création.

Dans la table « linfaune », les codes commençant par “ref” ont bien été définis comme clés étrangères.

The screenshot shows the pgAdmin III interface. On the left, the tree view shows the database structure, with the 'linfaune' table expanded and the 'ref_code_s' column selected. On the right, the 'Propriétés' (Properties) tab is active, displaying the column's attributes. The 'Clé étrangère' (Foreign key) property is set to 'Oui' (Yes), which is highlighted with a red box. Below the properties, the SQL script for the column is shown, including the 'ALTER TABLE' statements to add and modify the column.

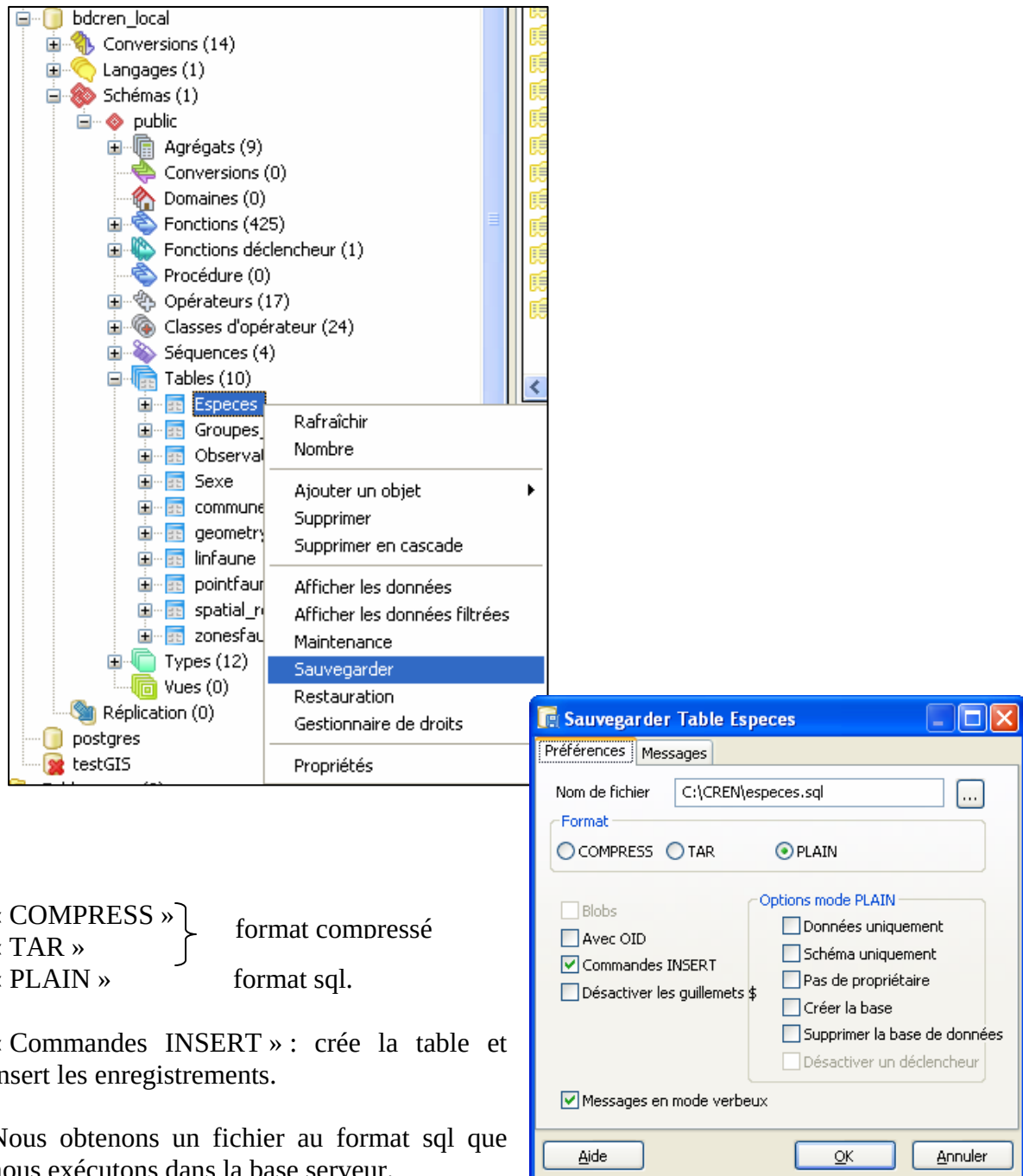
Propriété	Valeur
Nom	ref_code_s
Position	6
Type de données	numeric
Valeur par défaut	
Séquence	
Non NULL	Oui
Clé primaire	Non
Clé étrangère	Oui
Stockage	MAIN
Hérité(e)	Non
Statistiques	-1
Colonne système	Non
Commentaires	

```
-- Column: ref_code_s
-- ALTER TABLE linfaune DROP COLUMN ref_code_s;
ALTER TABLE linfaune ADD COLUMN ref_code_s numeric;
ALTER TABLE linfaune ALTER COLUMN ref_code_s SET STORAGE MAIN;
ALTER TABLE linfaune ALTER COLUMN ref_code_s SET NOT NULL;
```

2) Etape 2 : transfert de la base sur le serveur

a. PostgreSQL du poste local vers le serveur :

Nous devons importer les tables attributaire dans notre base située sur le serveur. Pour cela, il faut aller dans le menu contextuel de la table en local et sélectionner la commande « sauvegarder ».



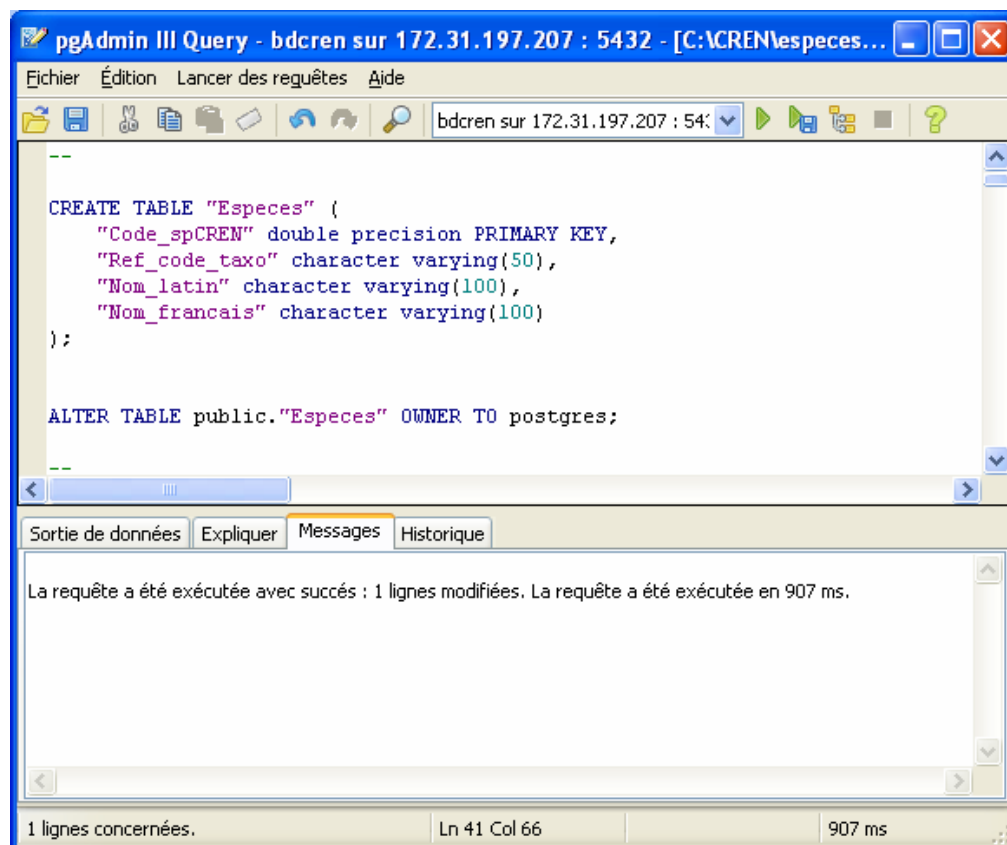
« COMPRESS » } format compressé
« TAR » }
« PLAIN » } format sql.

« Commandes INSERT » : crée la table et insert les enregistrements.

Nous obtenons un fichier au format sql que nous exécutons dans la base serveur.



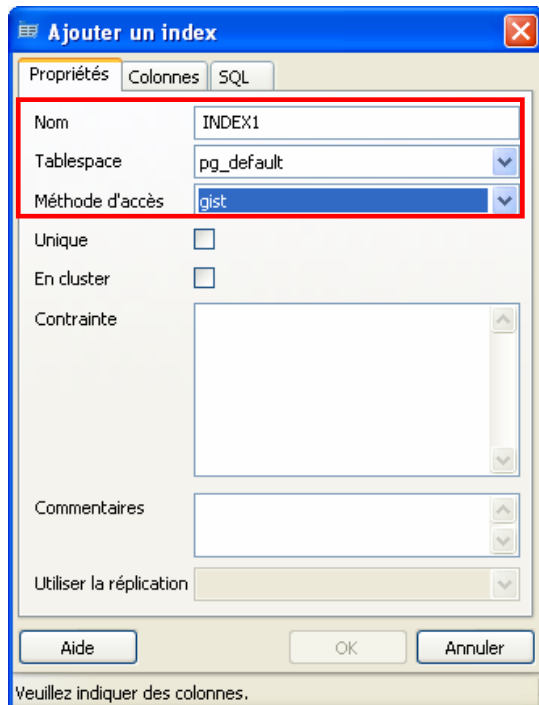
Ensuite, nous exécutons le fichier .sql dans le générateur de requête.



Ici, nous avons dû changer le propriétaire de la base. En effet, pour la base en local, le propriétaire est « sigma » alors qu'il s'agit de « postgres » sur le serveur. Nous faisons la même chose pour toutes les tables attributaires.

Dans l'onglet « Propriétés » :

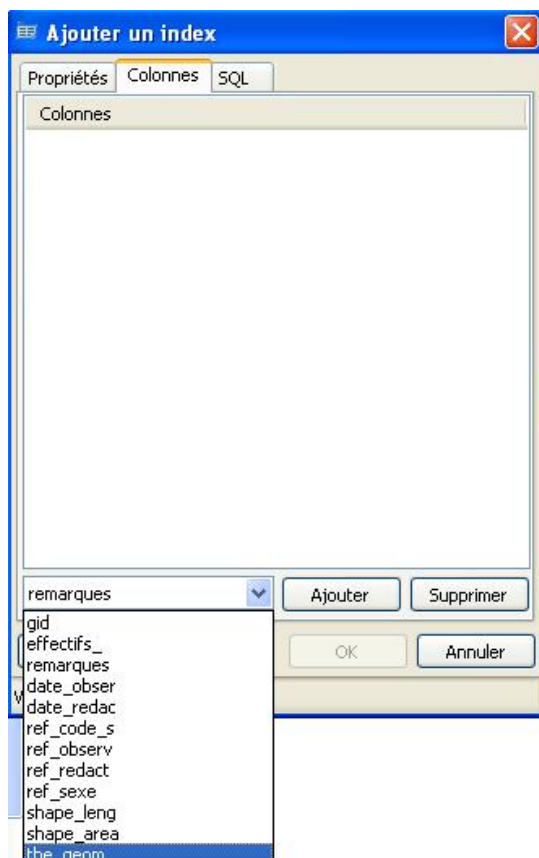
Nous sélectionnons « **Ajouter un objet/ Ajouter un index** », une fenêtre de paramétrage s'affiche :



The screenshot shows the 'Ajouter un index' dialog box with the 'Propriétés' tab selected. The 'Nom' field contains 'INDEX1', the 'Tablespace' dropdown is set to 'pg_default', and the 'Méthode d'accès' dropdown is set to 'gist'. These three fields are enclosed in a red rectangular box. Below these fields are checkboxes for 'Unique' and 'En cluster', both of which are unchecked. There is a large empty text area for 'Contrainte'. At the bottom, there is a 'Commentaires' text area and a 'Utiliser la réplication' dropdown menu. The 'OK' and 'Annuler' buttons are at the bottom right. A status bar at the bottom left says 'Veuillez indiquer des colonnes.'

Nous devons rentrer 3 paramètres :

- le nom de l'index (ici, index1)
- sélectionner l'option « pg_default » de la liste déroulante de « **tablespace** »
- sélectionner l'option « gist » de la liste déroulante de « **Méthode d'accès** »



The screenshot shows the 'Ajouter un index' dialog box with the 'Colonnes' tab selected. The 'Colonnes' list is empty. At the bottom, there is a dropdown menu labeled 'remarques' with a list of column names: gid, effectifs, remarques, date_observ, date_redac, ref_code_s, ref_observ, ref_redact, ref_sexe, shape_leng, shape_area, and the_geom. The 'Ajouter' button is highlighted. The 'OK' and 'Annuler' buttons are at the bottom right.

Dans l'onglet « Colonnes » :

Nous devons définir le champ à indexer dans cet onglet.

Pour cela, il faut sélectionner la colonne « **the_geom** » dans la liste déroulante (qui correspond toujours au champ géométrique) et cliquer sur « Ajouter ».

Nous validons par Ok.

Répétez la manipulation pour toutes les tables qui comportent un champ géométrique.

V. Passerelle d'exportation des données

A. *Export des données attributaires*

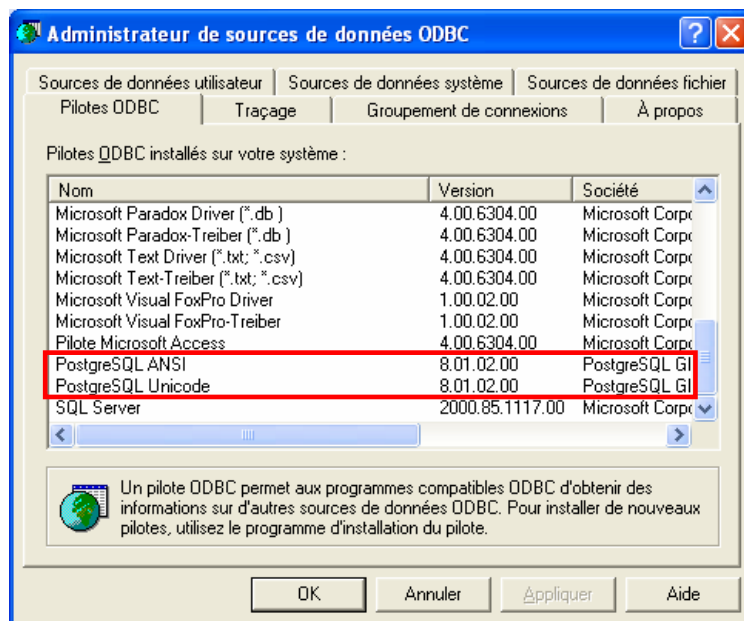
1) Le driver ODBC

Nous allons configurer la connexion entre notre base de données PostgreSQL et notre base Access grâce à un lien ODBC.

Le driver ODBC pour PostgreSQL a déjà été téléchargé et installé pour l'export des données à partir d'Access sur le site suivant :

<http://www.postgresql.org/ftp/odbc/versions/msi/>

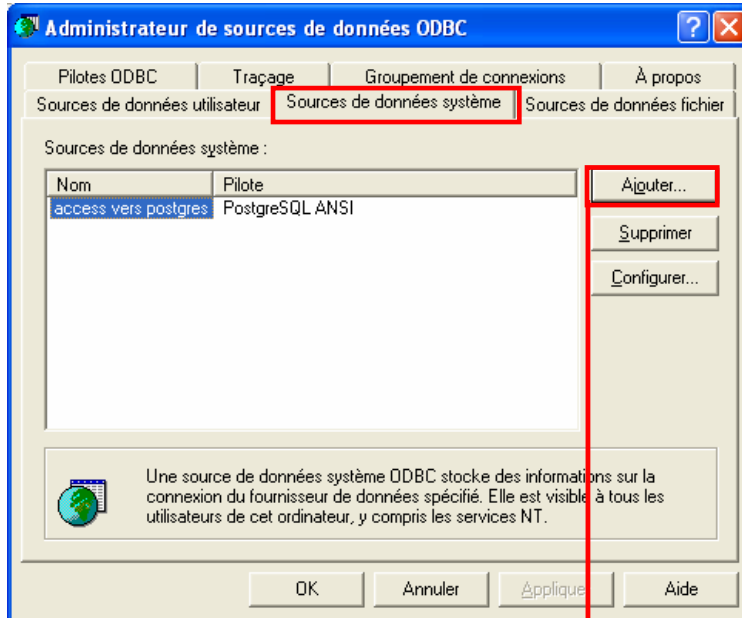
Le pilote se trouve bien dans la fenêtre « Administration de sources de données ODBC » accessible dans le « Panneau de configuration / Outils d'administration ».



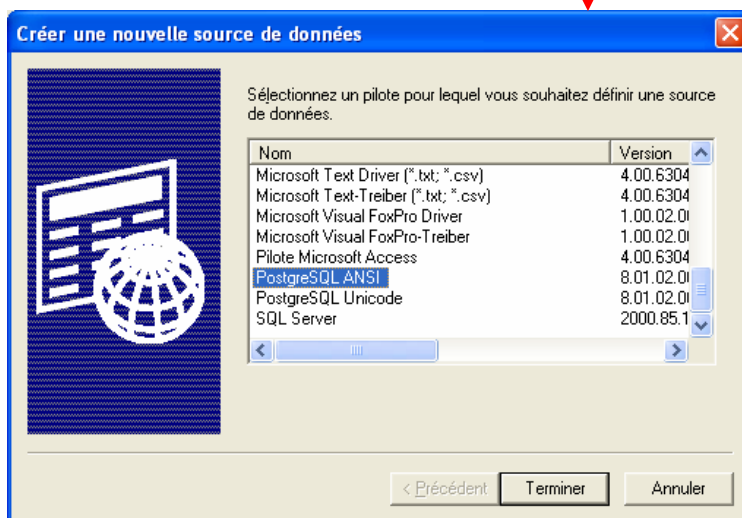
2) Configuration de la connexion

Nous allons transférer les données stockées sur PostgreSQL sur le serveur vers Microsoft Access sur le poste client.

Configuration du DSN PostgreSQL vers Access :



Nous choisissons de créer une connexion dans l'onglet « Sources de données système ».



Nous sélectionnons ici le driver « PostgreSQL ANSI ». C'est le pilote qui gère le mieux les caractères accentués sous Windows.

PostgreSQL ANSI ODBC

Data Source: Postgres vers access

Database: bdcren

Server: 172.31.197.45

User Name: postgres

Description:

SSL Mode: prefer

Port: 5432

Password: *****

Options: Datasource Global

Buttons: Save, Cancel, Datasource, Global

Paramètres de connexion avec la base de données « bdcren » sur le serveur.

Data Source : nom de la connexion.

Database : nom de notre base

Server : IP du serveur

Administrateur de sources de données ODBC

Pilotes ODBC | Traçage | Groupement de connexions | À propos

Sources de données utilisateur | Sources de données système | Sources de données fichier

Sources de données système :

Nom	Pilote
access vers postgres	PostgreSQL ANSI
Postgres vers access	PostgreSQL ANSI

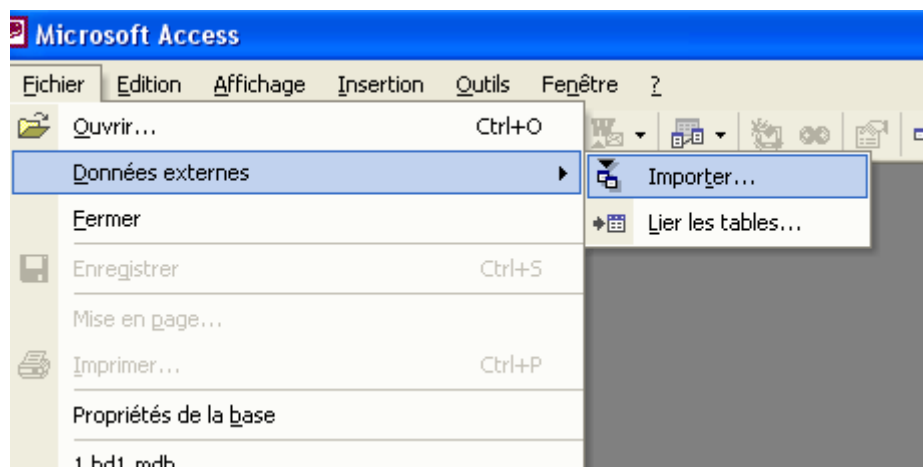
Buttons: Ajouter..., Supprimer, Configurer...

Une source de données système ODBC stocke des informations sur la connexion du fournisseur de données spécifié. Elle est visible à tous les utilisateurs de cet ordinateur, y compris les services NT.

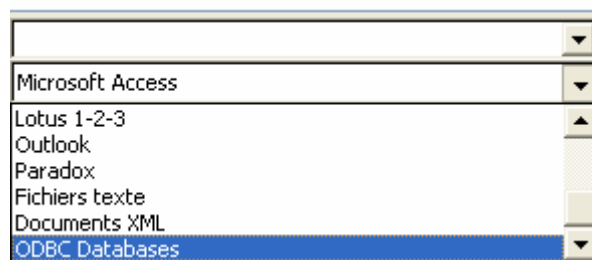
Buttons: OK, Annuler, Appliquer, Aide

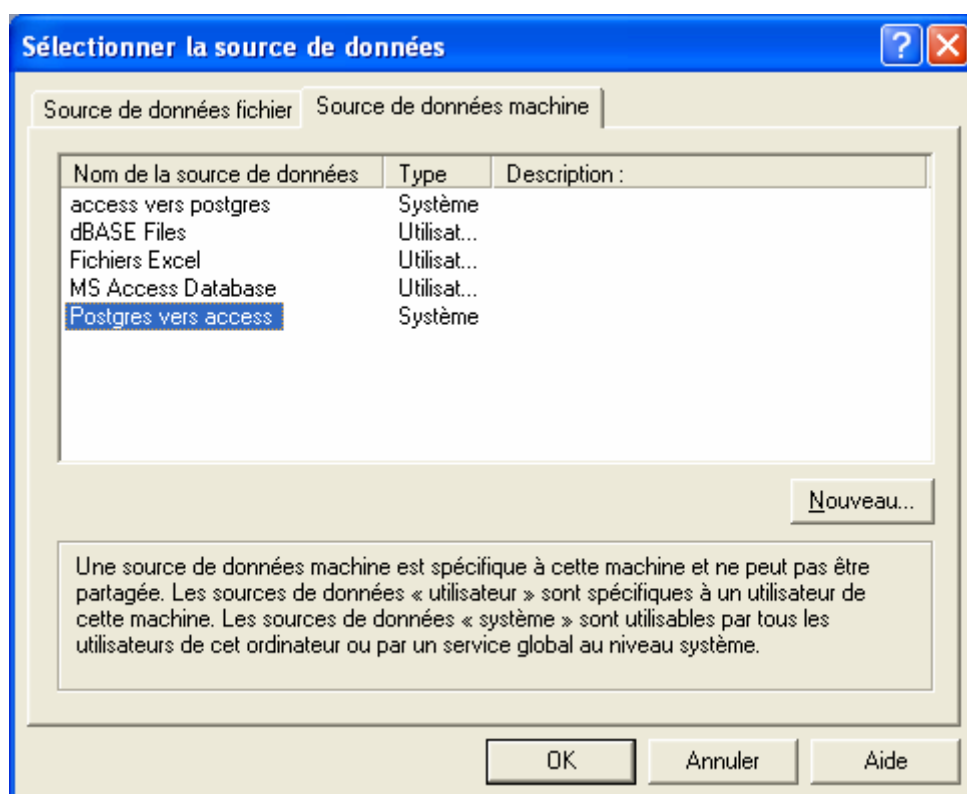
Notre nouvelle connexion s'est ajoutée dans l'onglet « Source de données système ».

Dans Access, nous importons les tables attributaires une par une en faisant :



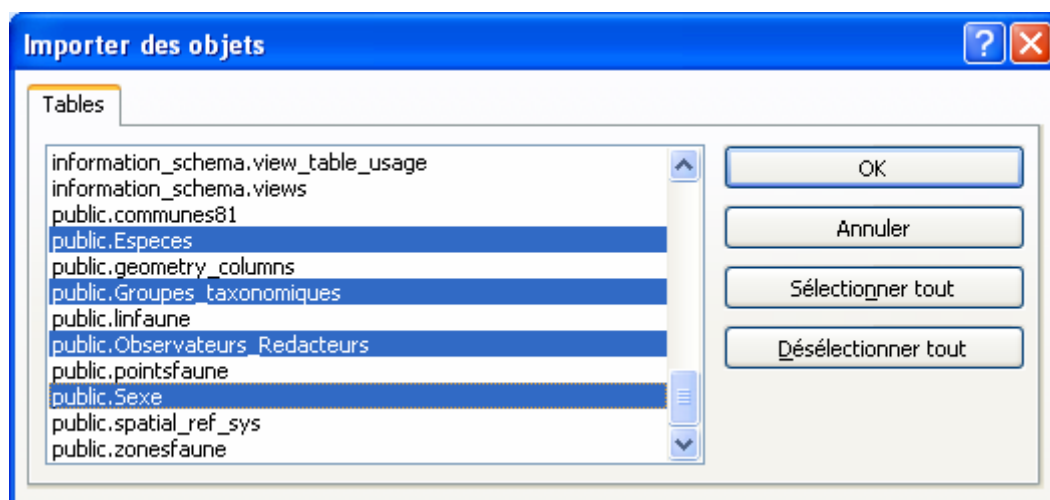
Nous choisissons ici notre type de fichier, à savoir « ODBC Databases () »



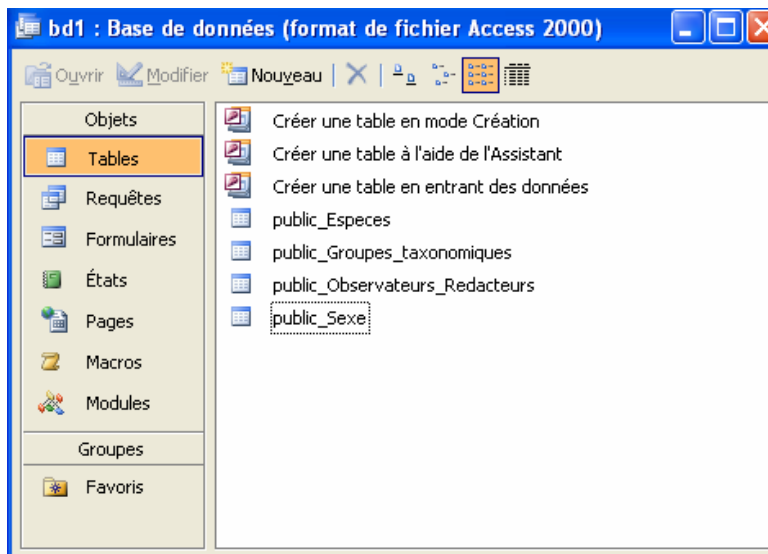


On nous demande de sélectionner la source de données, c'est-à-dire le lien ODBC que l'on vient de créer : « postgres vers access ».

Nous pouvons alors sélectionner les tables désirées à partir de l'utilitaire :



L'import s'effectue :



Les différentes tables ont été importées avec le préfixe « public ». Nous pouvons ensuite mettre à jours la base de données sous Access à partir de requête de mise à jour.

B. Export des données graphiques

Procédé : l'objectif est d'extraire de la base « bdcres » les données géométrique afin de créer un fichier shape. La création du fichier shape à partir de la base de données se fait par l'intermédiaire d'un utilitaire installé par défaut avec PostGIS : **pgsql2shp.exe**.

L'exportation vers un fichier shape sous PostGIS se fait donc en deux étapes :

- 1) La création d'un fichier shape avec **pgsql2shp.exe**.
- 2) Redéfinition du système de projection à partir de ArcCatalog.

Création du fichier Shape :

Cette étape se déroule sous l'invite de commande de Windows (Menu « Démarrer/programmes/Accessoires/Invite de commandes »).

Il faut se placer dans le dossier où vont être créés les fichiers Shape à intégrer. Les chemins d'accès ne sont spécifiés que pour l'IP du serveur et pour récupérer l'outil « **pgsql2shp.exe** ». La structure de la commande à taper pour exécuter le programme est de type :

Outil_de_conversion -h IP -u Nom_d'utilisateur -P mot_de_passe -g champ_geometrie Nom_base nomtable

Avec :

- **Outil_de_conversion** : L'utilitaire de conversion est cité plus haut (**pgsql2shp**). Il faut signifier l'emplacement de l'utilitaire **entre guillemet** qui est par défaut « C:\Program Files\PostgreSQL\8.1\bin\pgsql2shp.exe »
- **-h IP** : nous spécifions l'adresse IP du serveur.
- **-u Nom_d'utilisateur** : nous spécifions le nom de super-utilisateur de la base de données à attaquer.
- **-P mot_de_passe** : nous spécifions le mot de passe d'accès à la base de données à attaquer.
- **-g champ_geometrie** : nous spécifions le nom du champ contenant la géométrie.
- **Nom_base** : nom de la base de données.
- **nomtable** : nom de la nouvelle table à exporter.

Rq :

- Nous ne disposons pas de fonction d'export de la projection comme dans **shp2pgsql**. C'est pour cette raison que nous devons redéfinir le système de projection du fichier shape après sa création.
- Le nom du fichier shape n'est pas spécifié, il reprend le nom de la table exportée.
- Le shape est créé dans le répertoire courant.

Nous allons montrer en détail la création du fichier shape « zonesfaune » situé dans « c:/transfertserveurserveur »

Nous nous plaçons dans le répertoire voulu,

```
C:\> Invite de commandes
Microsoft Windows XP [version 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.

C:\Documents and Settings\Sigma>cd..
C:\Documents and Settings>cd..
C:\>cd transfertserveurserveur
C:\transfertserveurserveur>
```

Une fois placé dans le bon dossier il faut traduire la formule plus haut en l’adaptant à notre cas de figure :

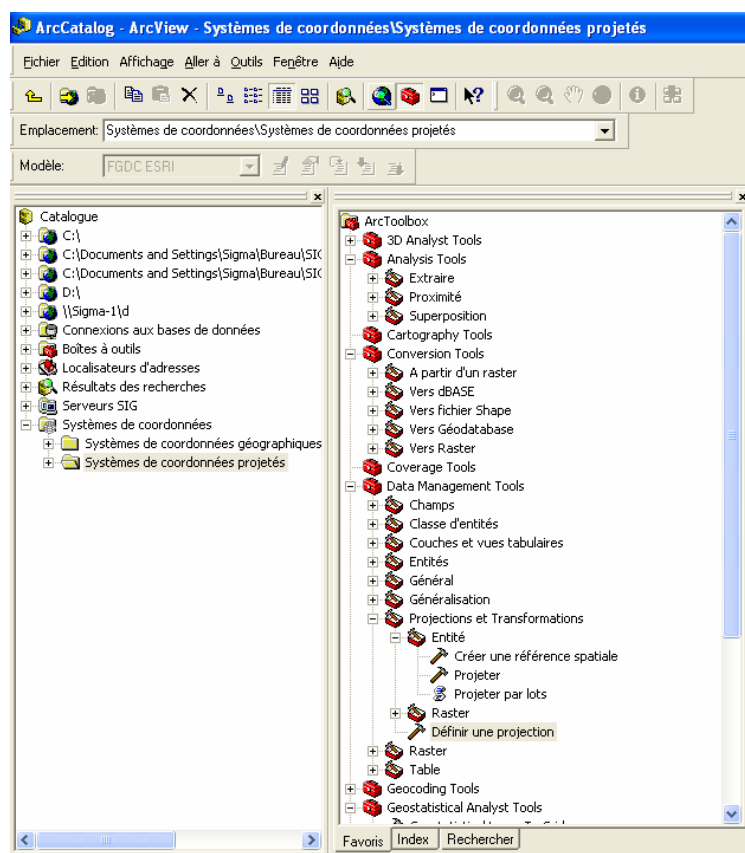
```
C:\transfertserveurserveur>"c:\Program Files\PostgreSQL\8.1\bin\pgsql2shp.exe" -h
172.31.197.207 -u postgres -P postgres -g the_geom bdcren zonesfaune
Initializing... Done (postgis major version: 1).
Output shape: Polygon
Dumping: XX [24 rows].
C:\transfertserveurserveur>
```

La commande lancée, nous obtenons le résultat suivant :

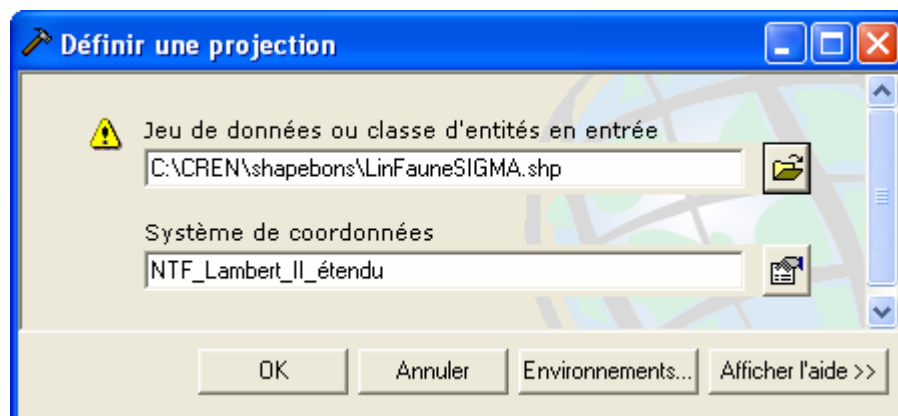
```
C:\> Invite de commandes
Microsoft Windows XP [version 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.

C:\Documents and Settings\Sigma>cd..
C:\Documents and Settings>cd..
C:\>cd transfertserveurserveur
C:\transfertserveurserveur>"c:\Program Files\PostgreSQL\8.1\bin\pgsql2shp.exe" -h
172.31.195.207 -u postgres -P postgres -g the_geom bdcren zonesfaune_
```

Le fichier est ouvert à partir d’ArcCatalog, et le système de projection est modifié à partir de l’ArcToolbox :



L’outil utilisé est « Définir une projection ». Nous remplissons la boîte de dialogue suivante :



Dans « Jeu de données ou classe d’entités en entrée », le nom du shape nouvellement créé est spécifié.

Dans « système de coordonnées », la projection utilisée pour nos données est indiquée. Il s’agit du Lambert II étendu.

La couche shape peut alors être utilisée.

ANNEXE 3

GUIDE D'UTILISATION DU LOGICIEL SIG

Sommaire

I.	GUIDE D'UTILISATION DE QUANTUM GIS	86
A.	INSTALLATION	86
B.	OUVERTURE DES DONNEES	87
1)	<i>Ouverture des données stockées en local.</i>	87
a.	Les données vectorielles	87
b.	Les données Raster	88
2)	<i>Ouverture des données stockées sur PostGIS sur un serveur distant.</i>	90
C.	VISUALISATION DES DONNEES	93
1)	<i>Propriétés d'affichage des couches</i>	93
a.	Les options d'affichage	93
b.	Les outils de navigation	95
2)	<i>Les données attributaires</i>	95
a.	Visualisation des données	95
b.	Edition des données vectorielles	97
D.	LES PRINCIPALES FONCTIONNALITES	102
1)	<i>Les requêtes</i>	102
2)	<i>Les analyses thématiques</i>	104
3)	<i>Les Plugins</i>	106
II.	GUIDE D'UTILISATION DE OPENJUMP	108
A.	L'INTERFACE	108
B.	IMPORT / EXPORT DE FICHIERS « SHAPE » (.SHP)	109
C.	CONSULTATION DES DONNEES ATTRIBUTAIRES - STRUCTURE DES TABLES	111
1)	<i>Visualisation des données</i>	111
2)	<i>Structure des tables</i>	112
D.	EDITION DES ENTITES GEOGRAPHIQUES	113
1)	<i>Sélection et déplacement d'entité</i>	114
2)	<i>Déplacement, ajout et suppression de sommet</i>	115
3)	<i>Saisie de lignes, points et polygones</i>	116
4)	<i>Lien données géographiques - données attributaires</i>	117
E.	LES REQUETES ET ANALYSES THEMATIQUES	118
1)	<i>Exemple d'une requête simple</i>	119
2)	<i>Exemple d'une analyse thématique</i>	120
F.	L'IMPORT DE FICHIERS RASTER	121
G.	FONCTIONNALITES SUPPLEMENTAIRES D'OPENJUMP	122
1)	<i>Options de sélection</i>	122
2)	<i>Options des couches</i>	122
3)	<i>Programmation</i>	123
4)	<i>Géotraitements</i>	123
5)	<i>Statistiques</i>	125
6)	<i>Editeur de formules</i>	125
7)	<i>Options des couches</i>	126

I. Guide d'utilisation de Quantum GIS



A. Installation

Quantum GIS est un SIG Open Source que l'on peut télécharger sur Internet à l'adresse suivante :

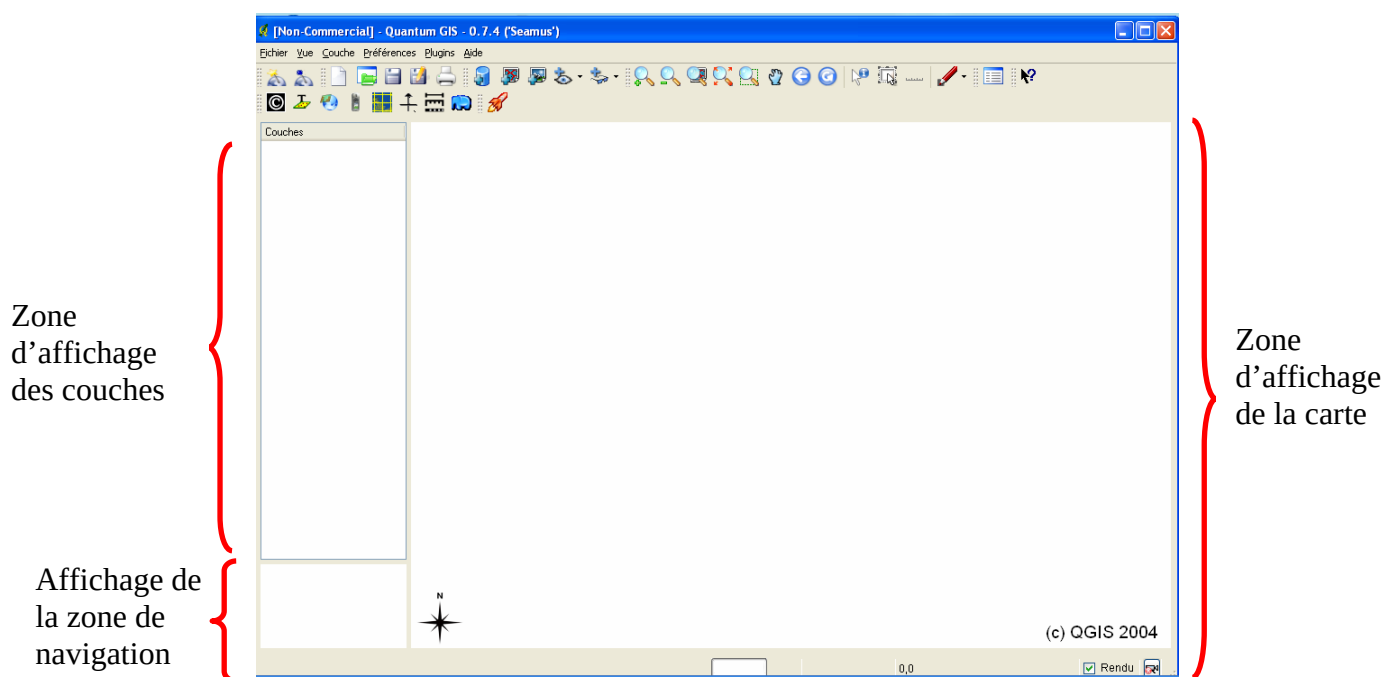
<http://qgis.org/>

Nous téléchargeons alors un fichier « .exe » qu'il faut exécuter pour démarrer l'installation. Celle-ci s'effectue sans difficulté particulière. Il suffit de se laisser guider par l'assistant.

Une fois le logiciel installé, nous le lançons dans le menu démarrer en cliquant sur l'icône



A l'ouverture, QGIS affiche une page vide et nous pouvons déjà appréhender l'organisation de l'interface



Nous allons dans ce guide d'utilisation décrire les manipulations à effectuer pour manipuler le logiciel aussi bien avec des données stockées en local ou par l'intermédiaire de notre serveur sur une base de données PostGIS.



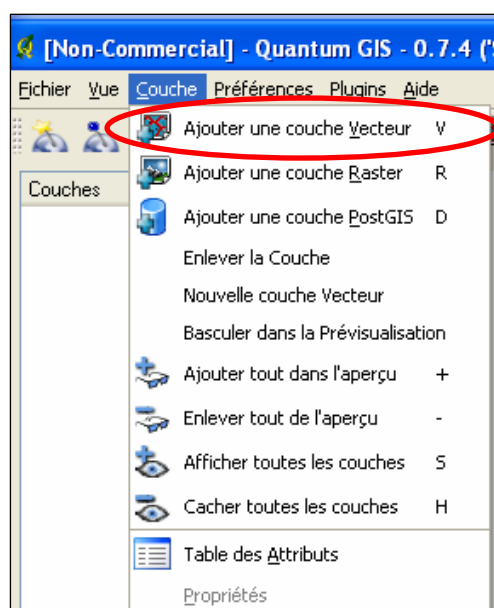
B. Ouverture des données

1) Ouverture des données stockées en local.

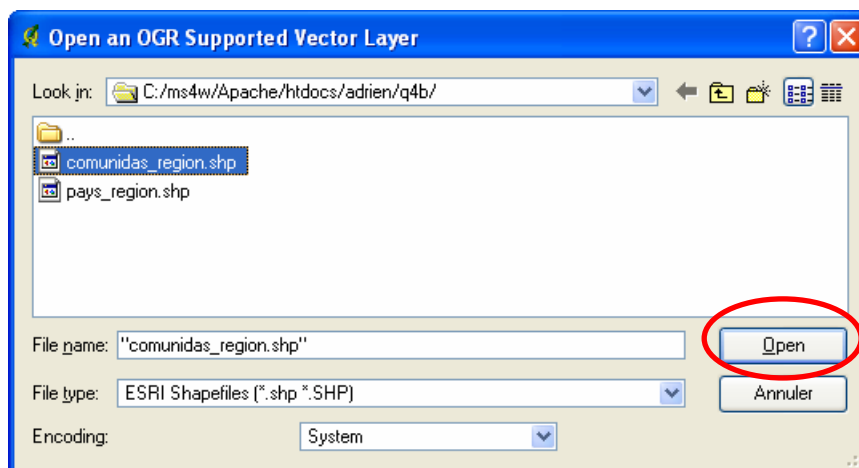
Le logiciel QGIS permet de traiter les deux types : vectoriel ou raster.

a. Les données vectorielles

Pour afficher une couche vectorielle, nous sélectionnons « **Couche / Ajouter une couche Vecteur** »

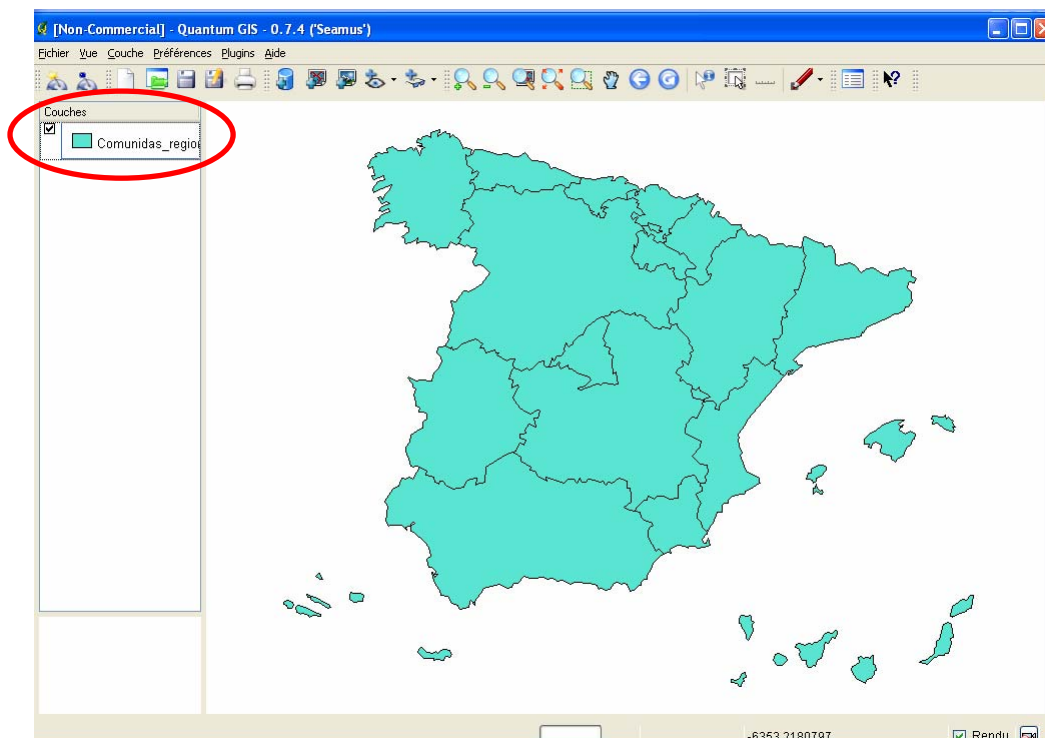


Ou cliquez directement dans la barre d'outils sur l'icône



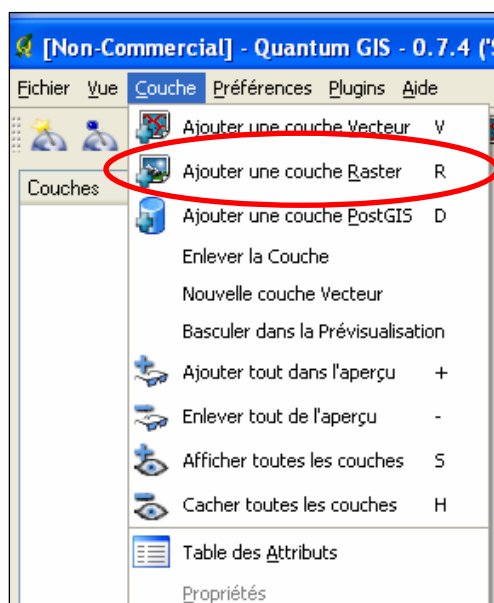
Nous sélectionnons ensuite la couche vectorielle que nous voulons afficher et validons en cliquant sur « **Open** »

La couche sélectionnée s'affiche alors dans le gestionnaire de couche et l'on peut visualiser la carte à l'écran :



b. Les données Raster

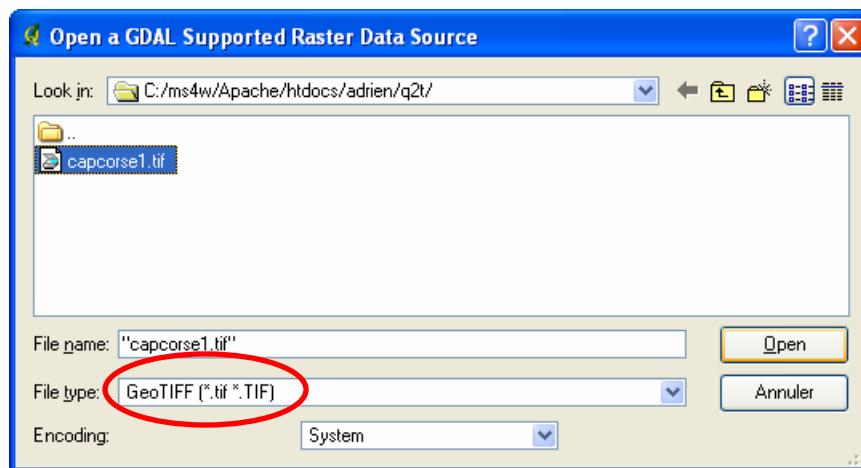
De la même manière que pour les données vectorielles, nous sélectionnons dans le menu « Couche / Ajouter une couche Raster ».



Nous pouvons aussi cliquer sur l'icône

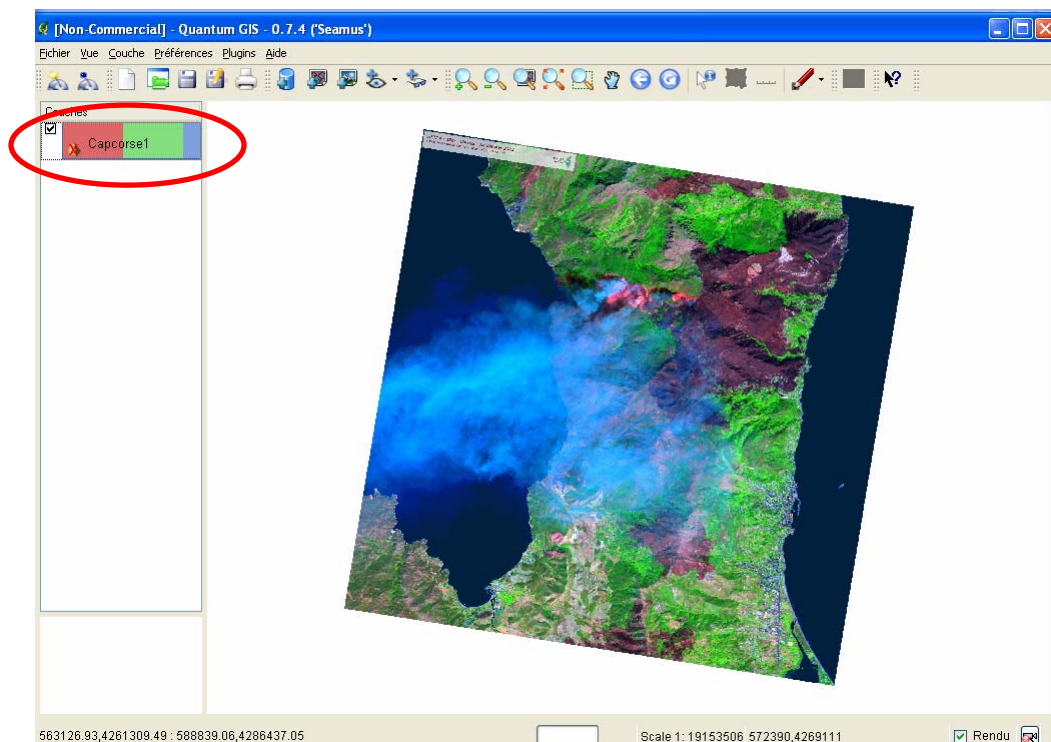


Dans la fenêtre d'ouverture, il faut sélectionner le type de raster que nous voulons intégrer



Il faut ensuite valider par « Ouvrir »

Le raster s'affiche alors dans la fenêtre de vue et de gestion des couches :

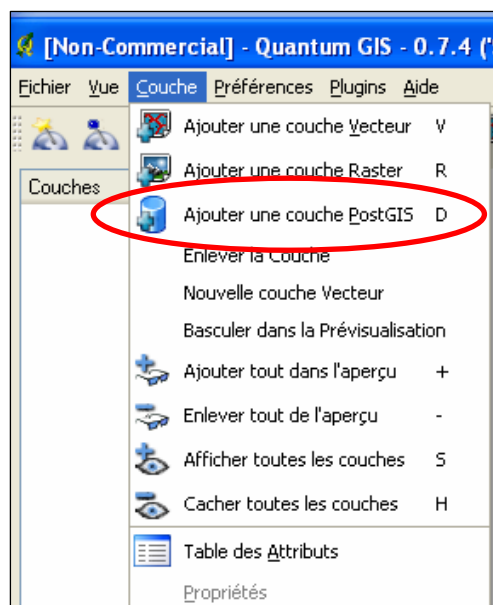


On peut constater que le type de couche est signalé dans le gestionnaire par un fond multicolore.

2) Ouverture des données stockées sur PostGIS sur un serveur distant

Quantum GIS est conçu pour pouvoir se connecter à une base de donnée distante. Il faut donc configurer la connexion pour pouvoir accéder à la base.

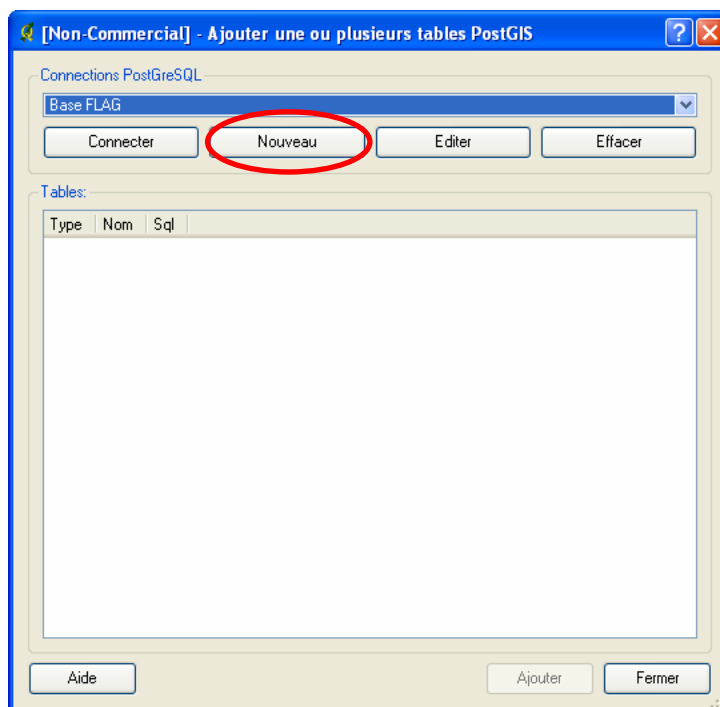
Nous sélectionnons dans le menu « Couche / Ajouter une couche PostGIS »



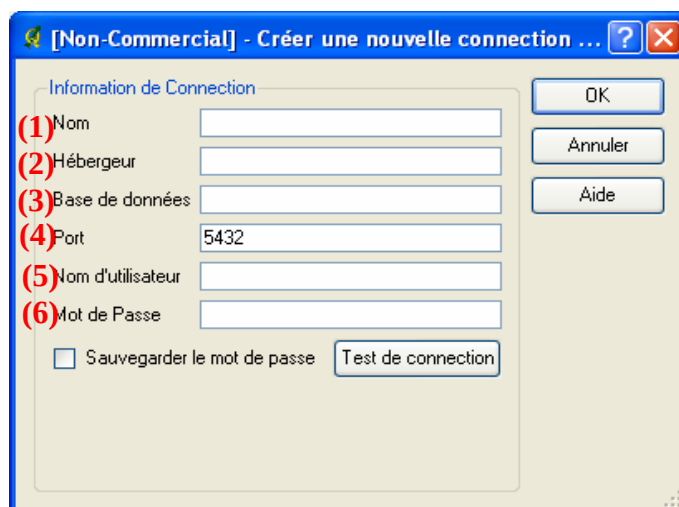
Il est aussi possible de sélectionner dans la barre d'outils, l'icône



Comme aucune connexion à la base n'a été encore définie, il va falloir en créer une nouvelle dans l'assistant d'ajout de tables PostGIS.



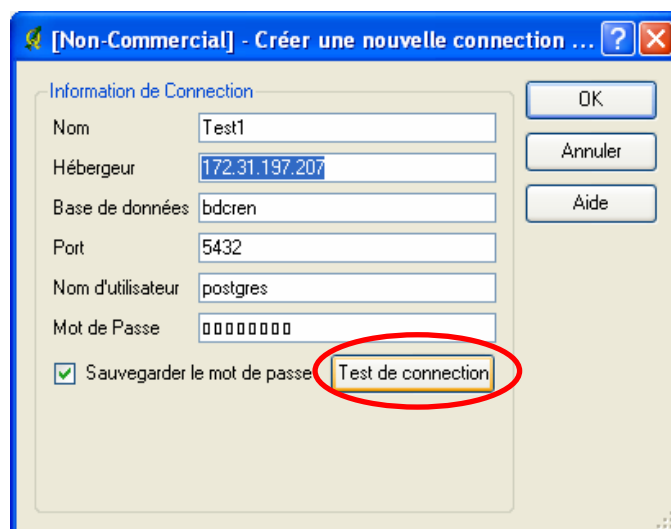
L'assistant de création de la connexion démarre alors.



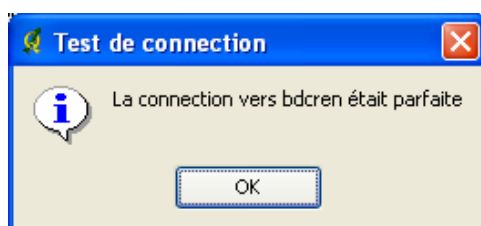
Nous devons entrer alors une série d'information concernant le serveur et la base de donnée :

- (1) En premier lieu il faut nommer la nouvelle connexion (ici test1)
- (2) Rentrer l'adresse IP du serveur dans la case « hébergeur »
- (3) Indiquer le nom de la base de données PostGIS recherchée (bdcren)
- (4) Le numéro de port est détecté par Windows XP
- (5) et (6) Enfin, saisir le nom d'utilisateur entré sur PostGIS et le mot de passe associé.

Nous obtenons la fenêtre suivante :

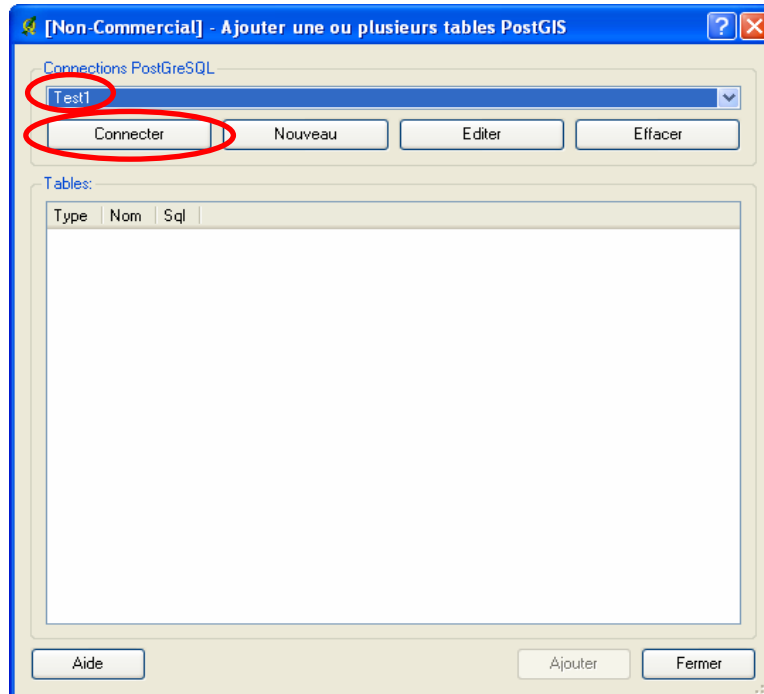


Il faut cliquer sur « Test de connexion » pour vérifier que PostGIS reconnaît bien la base de données externe.

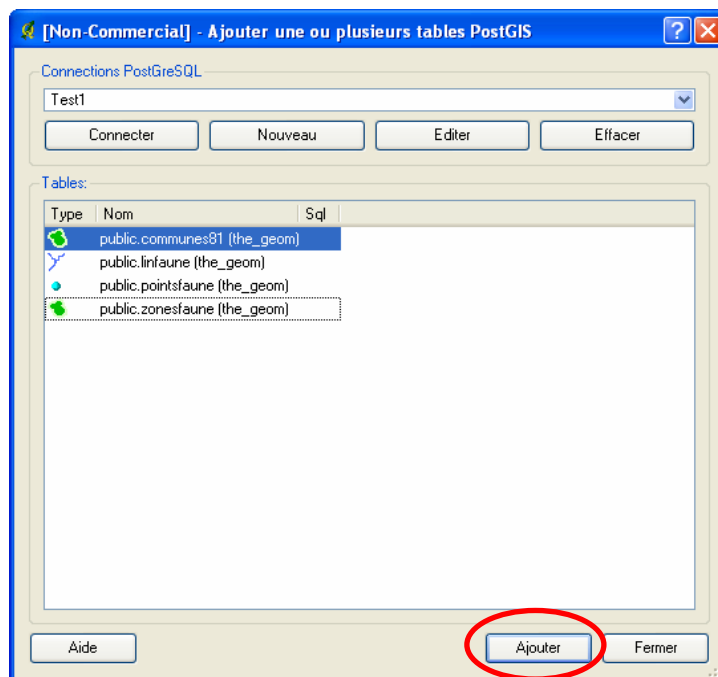


Le résultat du test est positif, la connexion fonctionne bien.

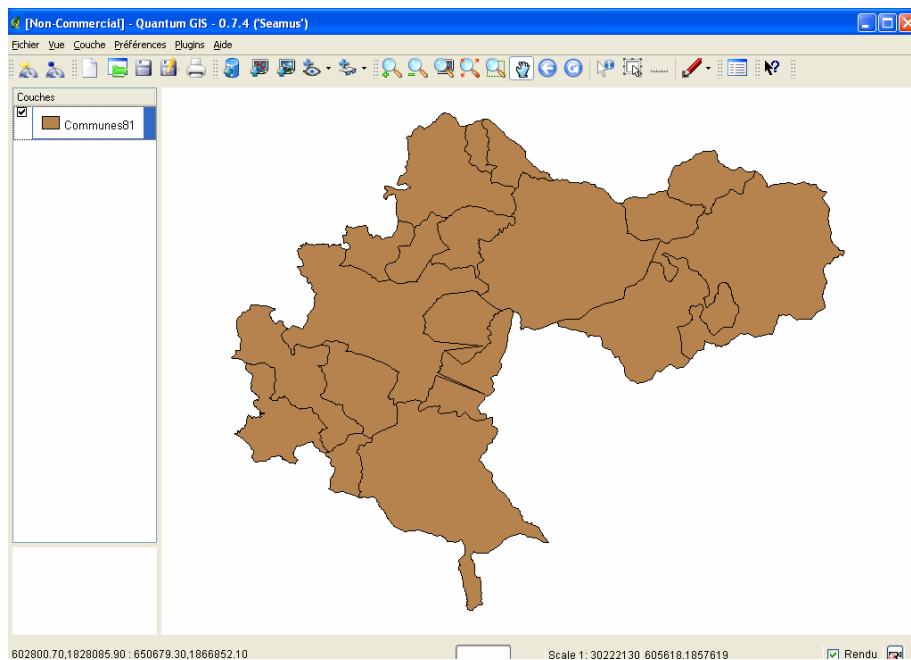
La liaison entre QGIS et notre base de données sur PostGIS. Pour accéder aux données de la base, nous devons sélectionner dans la liste déroulante la connexion que l'on vient de créer.



Nous cliquons ensuite sur « **connecter** » pour afficher les différentes couches disponibles



Nous sélectionnons la ou les couche(s) désirée(s) et cliquons sur « Ajouter ». La couche apparaît alors dans la carte.



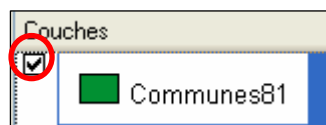
C. Visualisation des données

1) Propriétés d'affichage des couches

a. Les options d'affichage

- Affichage des couches vectorielles

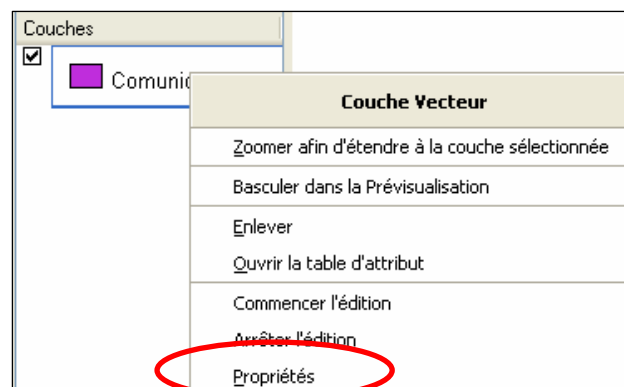
Pour afficher une couche ou masquer une couche, nous utilisons la case associée dans le gestionnaire des couches :



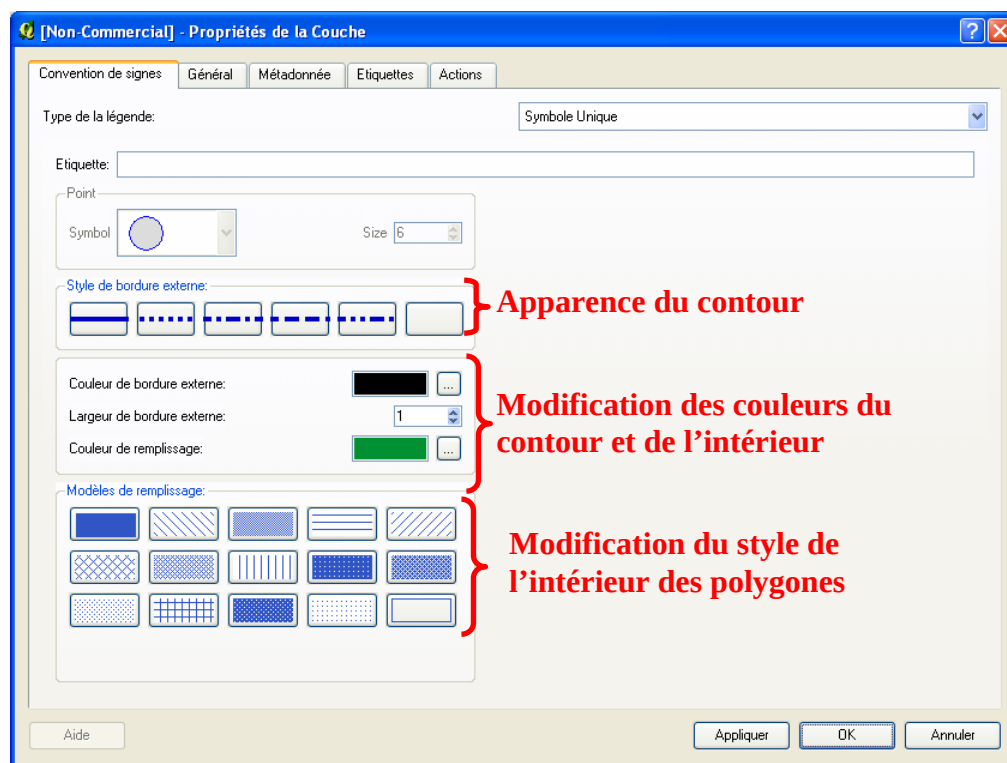
Nous pouvons afficher ou masquer l'ensemble des couches en cliquant sur l'icône



Les caractéristiques d'affichage d'une couche peuvent être modifiées en faisant un clic droit sur la couche à modifier et en sélectionnant les « Propriétés » :



Une fenêtre de configuration s'ouvre alors où nous pouvons modifier l'aspect de l'affichage :

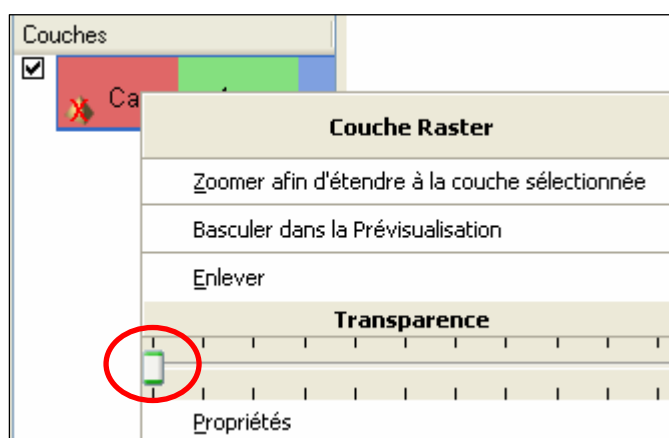


Vous pouvez donc modifier de manière simple l'apparence de chaque couche.

- Affichage des couches raster

Les possibilités d'affichages des couches raster sont plus limitées. Nous pouvons, si nous le souhaitons, jouer sur la transparence de la couche. Cela peut être utile lors de la superposition de couches vectorielles.

Pour cela, nous accédons au menu contextuel de la couche (clic droit) :



Il suffit de déplacer le curseur sur la barre pour jouer sur la transparence.

b. Les outils de navigation

La barre d'outil de QGIS contient différents outils pour modifier la visualisation de la carte.



Ces icônes permettent d'effectuer un zoom ou un de



Cette icône permet d'agrandir la vue.



Cliquez sur cette icône pour afficher l'extension maximale des couches.



Cette icône permet de zoomer sur les objets sélectionnés dans la couche



La main permet de déplacer la carte manuellement



Cliquez sur cette icône pour revenir à la vue précédente

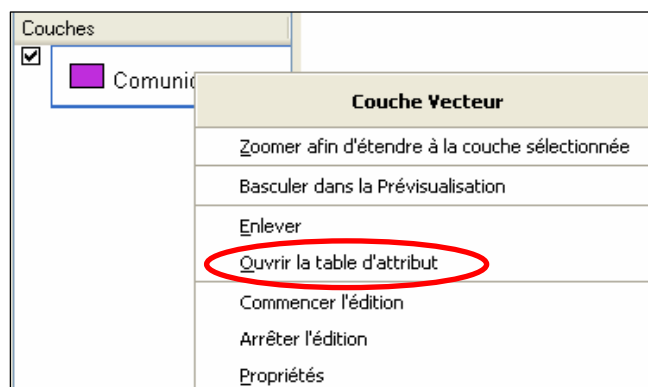


Enfin cette icône permet d'actualiser la vue en cours

2) Les données attributaires

a. Visualisation des données

Pour afficher les données attributaires d'une couche vectorielle, nous faisons un clic droit sur la couche.



Nous sélectionnons ensuite « Ouvrir la table d'attribut ».

Il est aussi possible de cliquer sur l'icône  dans la barre d'outils.

La table des attributs s'ouvre :

	perimeter	idign	insee	nom	cant	arr
1	0.000036043.2000000	310000312.000	81314	VIANE		15
2	0.000011425.4500000	310000280.000	81282	SENAUX		15
3	0.000018601.6800000	310000084.000	81085	ESCROUX		15
4	0.000018684.2000000	310000023.000	81023	BARRE		22
5	0.000039678.2700000	310000191.000	81192	MURAT-SUR-V		22
6	0.000015872.2700000	310000123.000	81124	LACAUNE		15
7	0.000017003.2800000	310000187.000	81188	MOULIN-MAGE		22
8	0.000022582.6900000	310000102.000	81103	GIJOUNET		15
9	0.000016869.0900000	310000028.000	81028	BERLATS		15
10	0.000020337.4900000	310000085.000	81086	ESPERAUSSE		15
11	0.000038996.4700000	310000061.000	81062	CASTELNAU-I		04
12	0.000019939.0900000	310000192.000	81193	NAGES		22
13	0.000017700.2100000	310000152.000	81153	LE MARGNES		04
14	0.000011269.0400000	310000031.000	81031	LE BEZ		04
15	0.000029049.6900000	310000133.000	81134	LAMONTELAR		03
16	0.000024330.0700000	310000037.000	81037	BRASSAC		04
17	0.000032177.0000000	310000053.000	81053	CAMBOUNES		04
18	0.000036449.7600000	310000014.000	81014	ANGLES		03

Commencer l'édition Arrêter l'édition Fermer



Cette icône permet de désélectionner toutes les entités



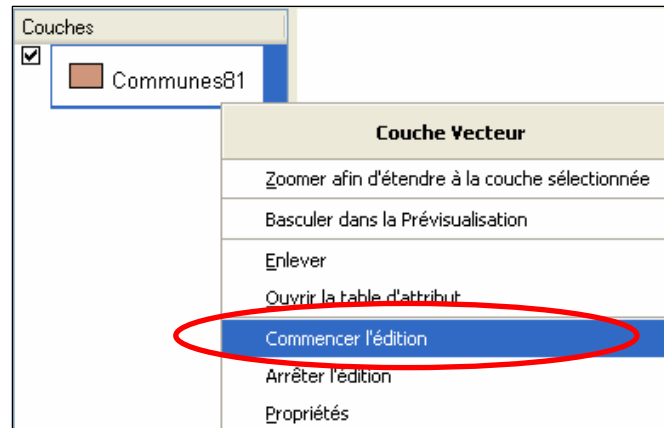
Cliquez sur cette icône pour placer la sélection en haut de la table



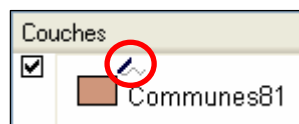
Cliquez sur cette icône pour inverser la sélection.

b. Edition des données vectorielles

Par défaut, PostGIS est dans un mode qui permet la simple visualisation des couches. Pour éditer une couche vectorielle, il faut passer en mode « Edition ».



Nous faisons un clic droit sur la couche à modifier et sélectionnons « Commencer l'édition »



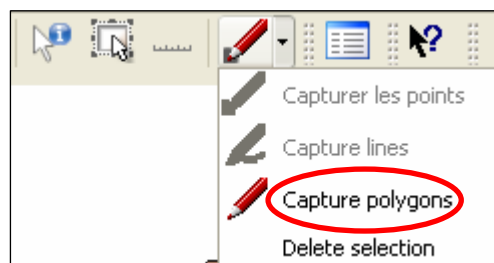
Un petit sigle vient signaler le changement de mode pour la couche

Les diverses opérations d'édérations sont disponibles en cliquant sur l'icône 

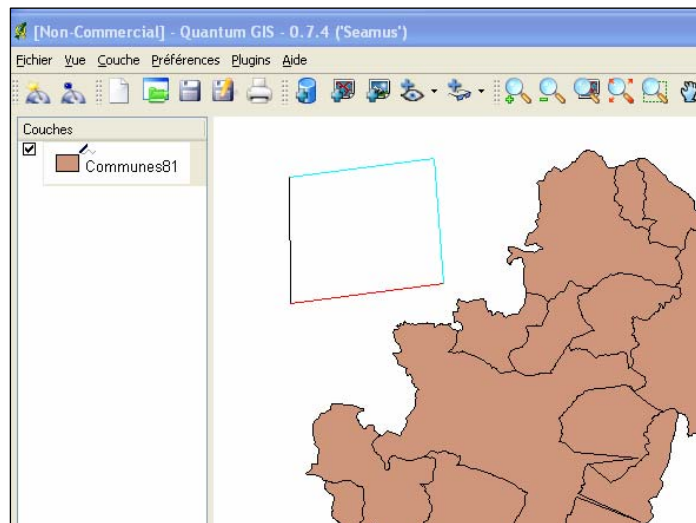
Ajout d'une entité vectorielle

L'ajout d'entité se fait par la même manipulation quelque soit le type de couche vectorielle. Nous ajouterons ici un polygone dans la couche des communes stockée sous PostGIS.

Pour ajouter une entité, nous développons l'icône édition et sélectionnons « Capture Polygons »



Le curseur devient alors une cible et nous pouvons dessiner le polygone sur la carte :



Lorsque celui-ci est tracé, nous faisons un clic droit afin de valider la fin de la saisie. Une petite fenêtre s'affiche alors automatiquement.

Elle va nous permettre de renseigner les divers attributs de l'objet que nous venons de créer

	field	value
1	gid	NULL
2	area	NULL
3	perimeter	NULL
4	idign	NULL
5	insee	NULL
6	nom	NULL
7	cant	NULL
8	arr	NULL
9	reg	NULL
10	shape_leng	NULL

OK Annuler

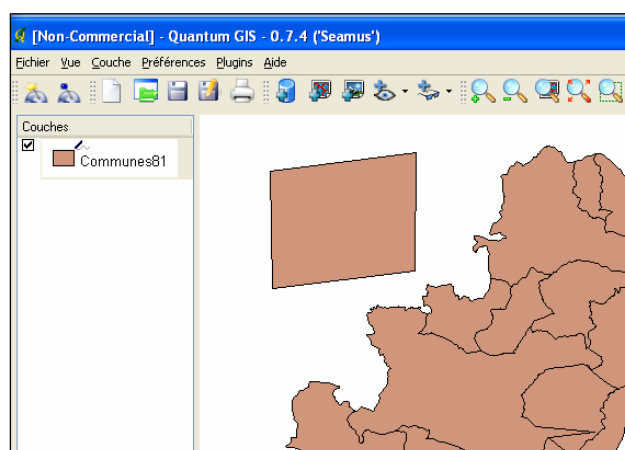
Nous renseignons alors tous les champs que nous souhaitons en prenant soins de respecter leur type.



	field	value
1	gid	99999
2	area	99999
3	perimeter	99999
4	idign	99999
5	insee	99999
6	nom	test creation
7	cant	99999
8	arr	99999
9	reg	99999
10	shape_leng	NULL
11	shape_area	NULL

OK Annuler

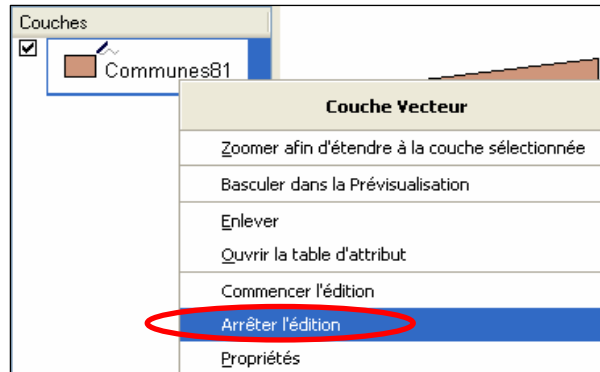
Nous cliquons sur « Ok » pour valider la saisie. Le polygone a bien été créé et l'on peut le voir s'afficher dans la même apparence que les autres objets de la couche.



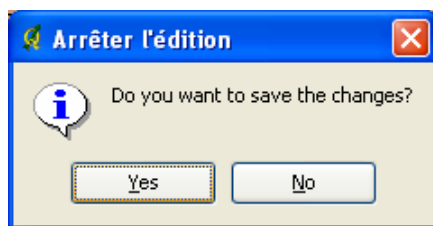
Lorsque nous avons effectué toutes les saisies nécessaires, nous devons veiller à bien sauvegarder les modifications.

Le seul moyen d'enregistrer les changements est de quitter le mode d'édition.

Pour cela, il faut faire un clic droit sur la couche qui vient d'être modifiée dans le gestionnaire.



Une fenêtre s'affiche alors et nous demande si l'on veut sauvegarder les changements



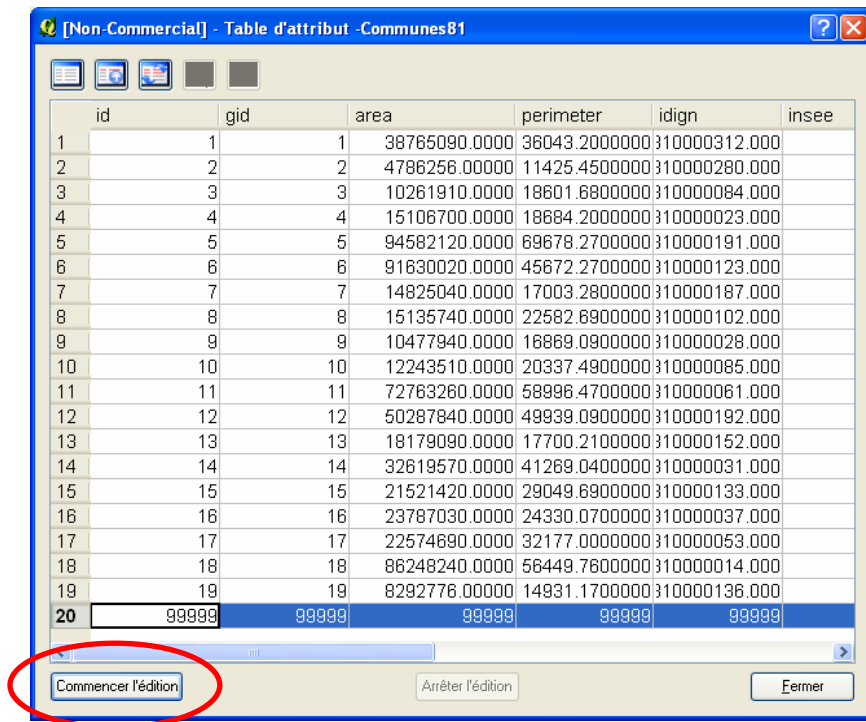
Nous cliquons sur « Yes » pour valider.

Si nous éditons une couche PostGIS, nous devons bien penser à vérifier sous PgAdmin que l'objet a été rajouté. Nous pouvons constater en revanche qu'il ne se rajoute pas automatiquement dans PostGIS, il faut pour cela recharger la couche.

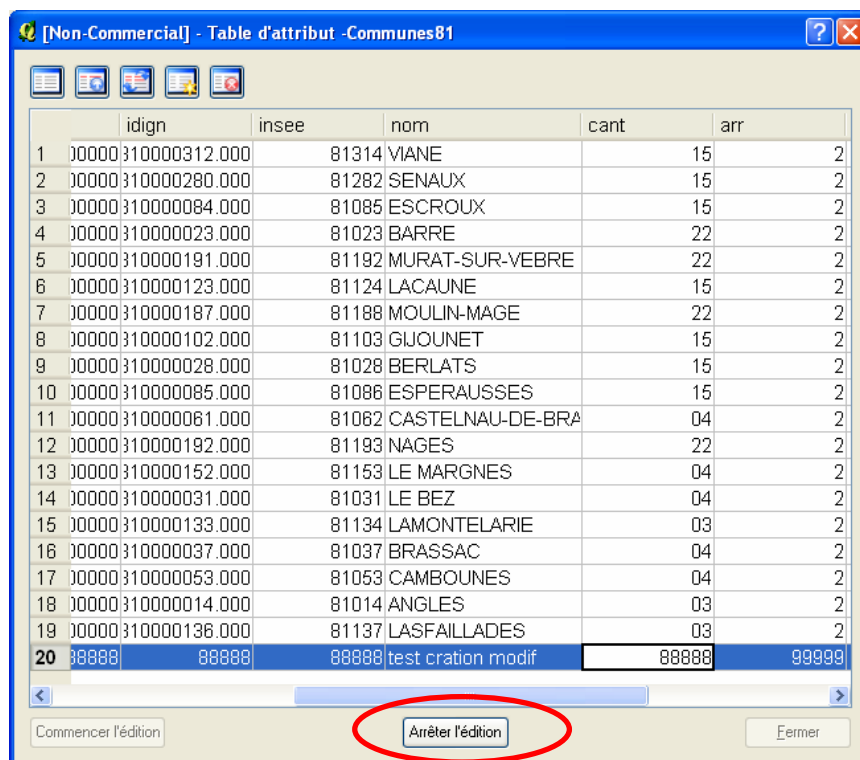
Modification d'une entité vectorielle

Si QGIS permet l'ajout et la suppression d'entités géométriques, nous ne pouvons en revanche pas modifier la géométrie d'une entité existante. Les seules modifications possibles sur une entité s'opèrent sur les données attributaires.

Pour cela, nous ouvrons la table de données de la couche à modifier. Nous allons ici modifier le polygone que nous venons de créer.



Nous devons alors démarrer le mode d'édition en cliquant sur « commencer l'édition ». Puis, nous cliquons sur la case à modifier et saisissons la nouvelle valeur.



Nous avons remplacé les 9 par des 8.

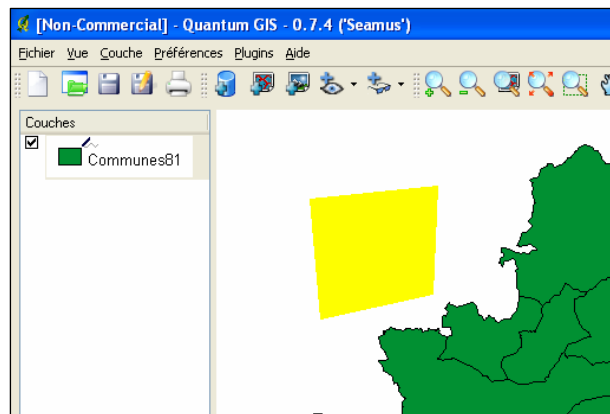
Nous cliquons enfin sur « Arrêter l'édition » et sauvegardons les modifications.

Nous pourrions vérifier que les modifications ont été bien prises en compte par PostGreSQL.

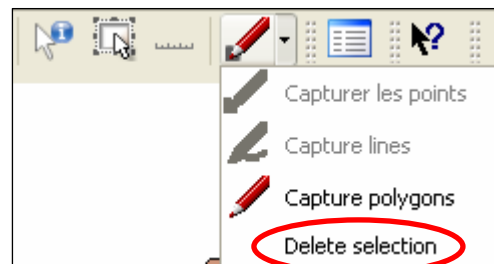
Suppression d'une entité

Pour supprimer une entité, nous sélectionnons la couche souhaitée et démarrons le mode édition.

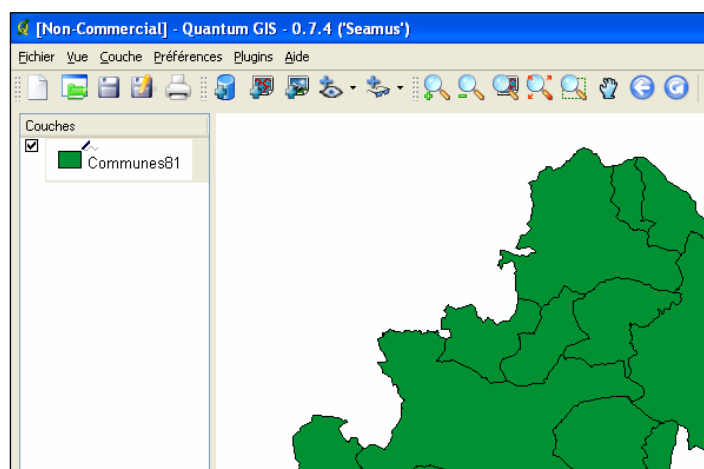
Nous sélectionnons ensuite l'entité à supprimer. Supprimons ici le polygone modifié précédemment.



Nous déroulons l'icône d'édition dans la barre d'outils et cliquons sur « Delete Selection »



Le polygone est alors supprimé et disparaît de la carte :



Nous devons sauvegarder la suppression en sortant du mode d'édition.

Pour que les changements soient pris en compte dans la table attributaire de QGIS, nous rechargeons la couche. Nous pouvons vérifier que la couche a bien été supprimée dans PostGIS.



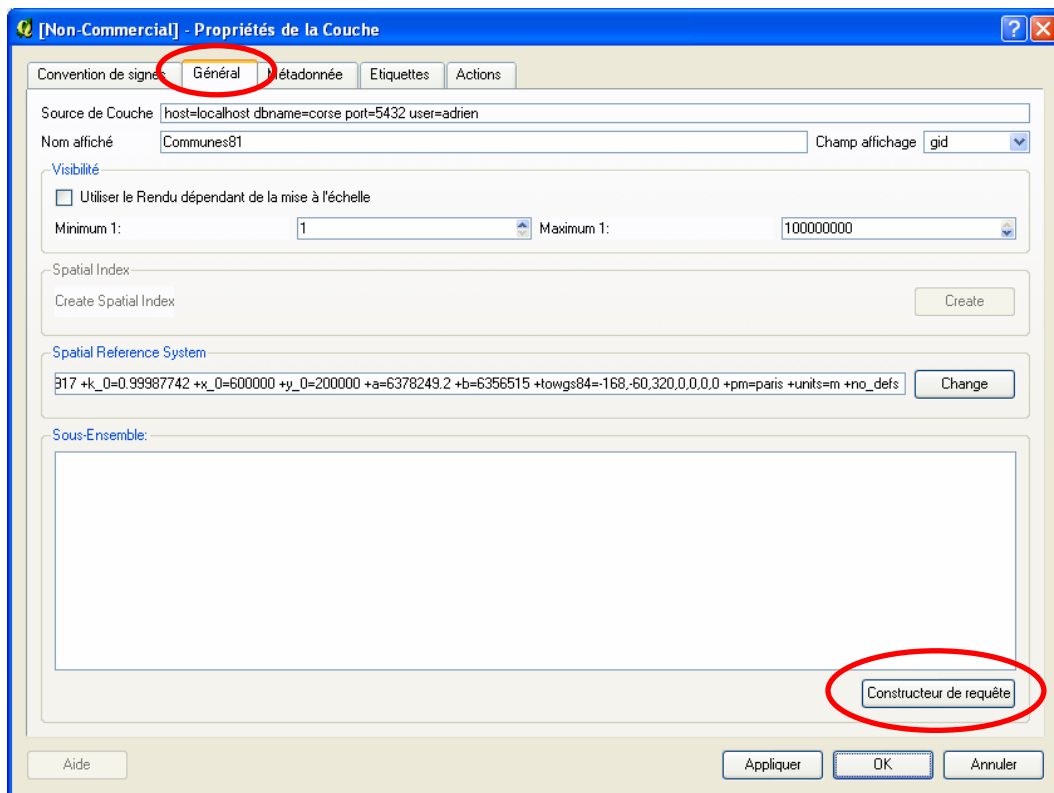
D. Les principales fonctionnalités

1) Les requêtes

QGIS permet d'effectuer des requêtes portant sur les attributs d'une couche. Ce ne sont pas des requêtes classiques de sélection mais des requêtes d'affichage.

Ainsi, seuls les résultats de la requête seront affichés.

Pour accéder au générateur de requête, nous lançons les propriétés de la couche. En guise d'exemple, nous effectuerons une requête portant sur la surface des communes.



Nous nous plaçons dans l'onglet « Général » et démarrons le constructeur de requêtes.

Sélection du champ sur lequel porte la requête

Opérateurs logiques

Expression de la requête

Echantillon de données

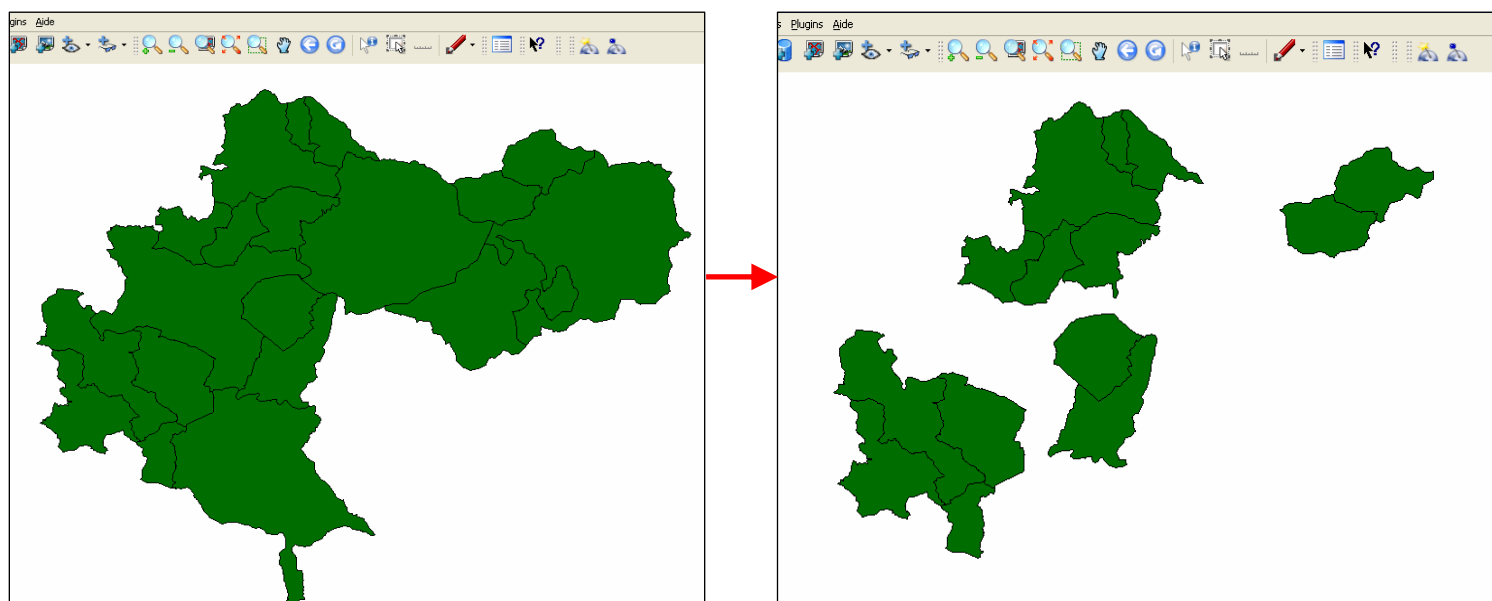
Nous chercherons ici à afficher les communes dont le périmètre est inférieur à 45672 m. Nous sélectionnons d'abord le champ concerné par la requête et cliquons sur « Echantillon » pour avoir un aperçu de ses valeurs. Un double clic sur un champ, un opérateur ou une valeur renseigne directement l'expression de la requête.

Il faut saisir ici « perimeter < 45672 »

Nous validons par Ok pour revenir aux propriétés de la couche.

Nous constatons que le critère de recherche a été ajouté dans les propriétés d'affichages

Voici le résultat final de la requête, nous passons d'une couche où toutes les communes sont affichées à une couche qui présente seulement les objets correspondant aux critères.

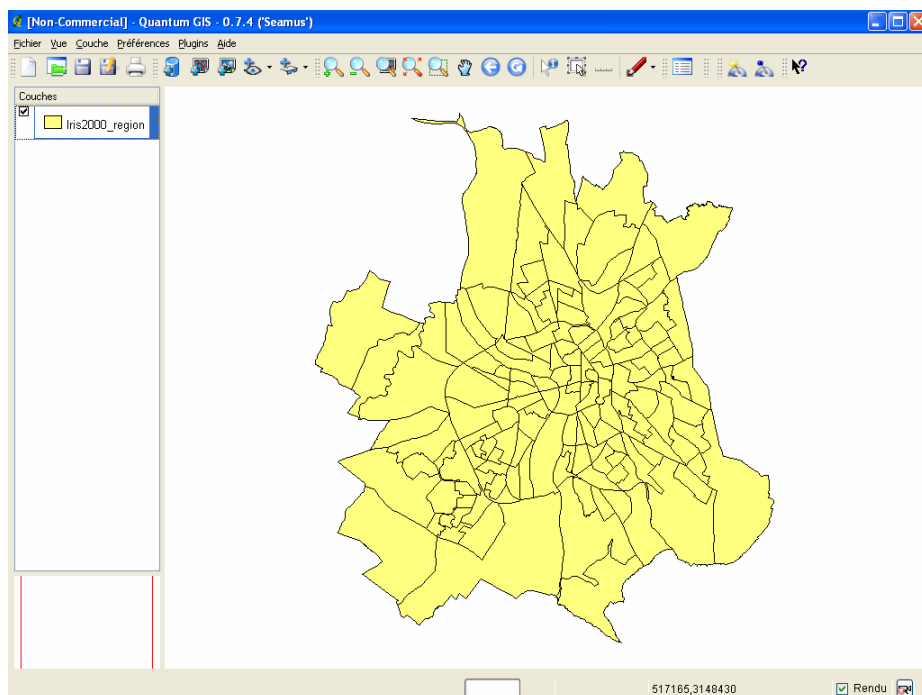


Pour retourner à un affichage total de la couche il faut supprimer les critères de recherche dans l'onglet « Général » des propriétés d'affichage.

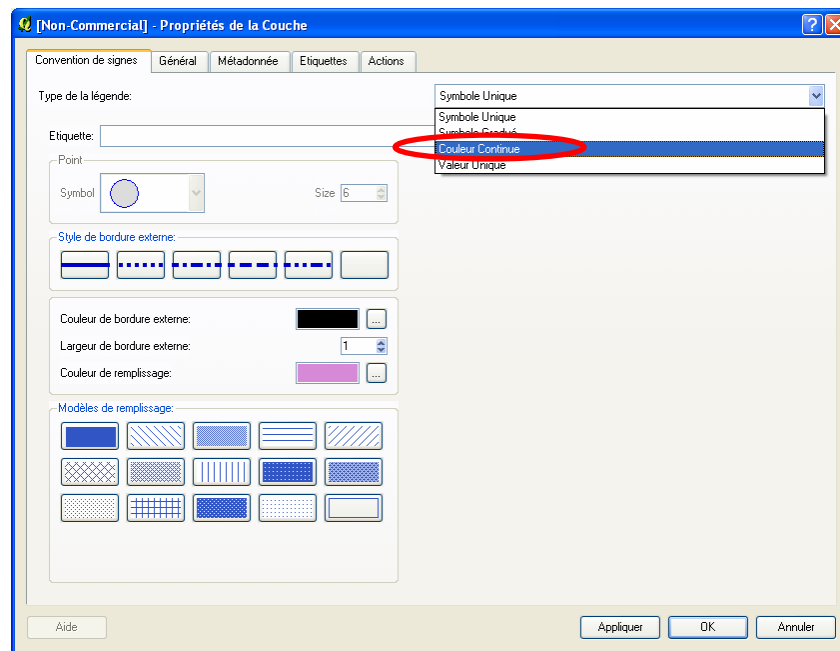
2) Les analyses thématiques

QGIS permet aussi d'effectuer des analyses thématiques sur tous les types de couches vectorielles.

Nous allons ici effectuer une analyse thématique sur la densité de population des îlots IRIS dans l'agglomération toulousaine en 1999. La couche vectorielle est la suivante :

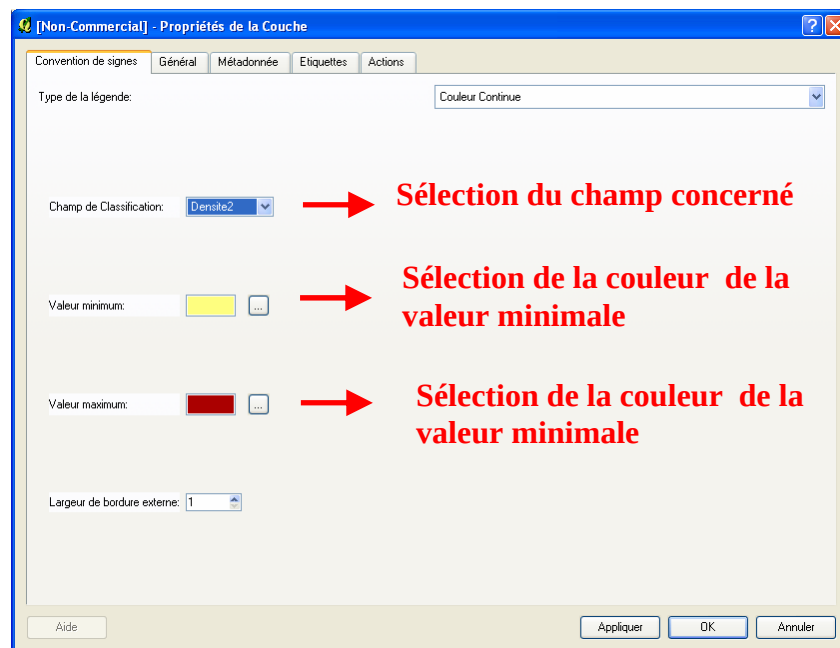


Pour effectuer une analyse thématique, nous ouvrons l'onglet « Convention des signes » des propriétés d'affichage.



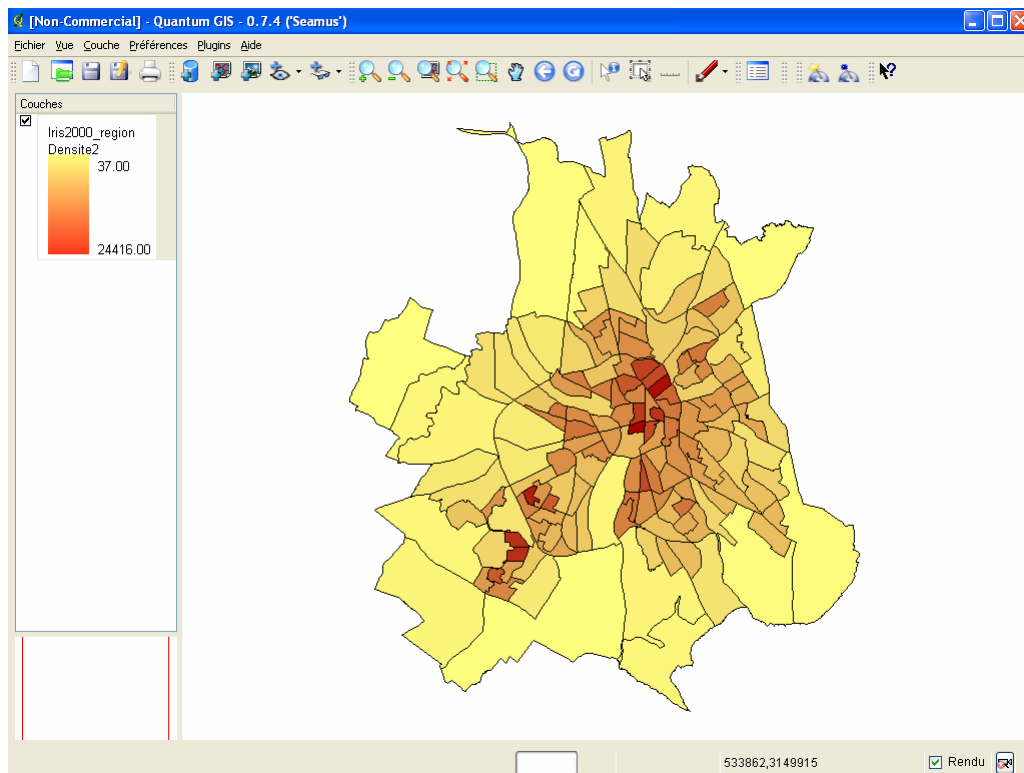
Dans la liste déroulante du type de légende, nous sélectionnons « Couleur continue » pour afficher une analyse thématique en page de couleur.

La fenêtre change d'apparence pour afficher les propriétés de l'analyse thématique :



Une fois l'analyse thématique paramétrée, nous validons par Ok.

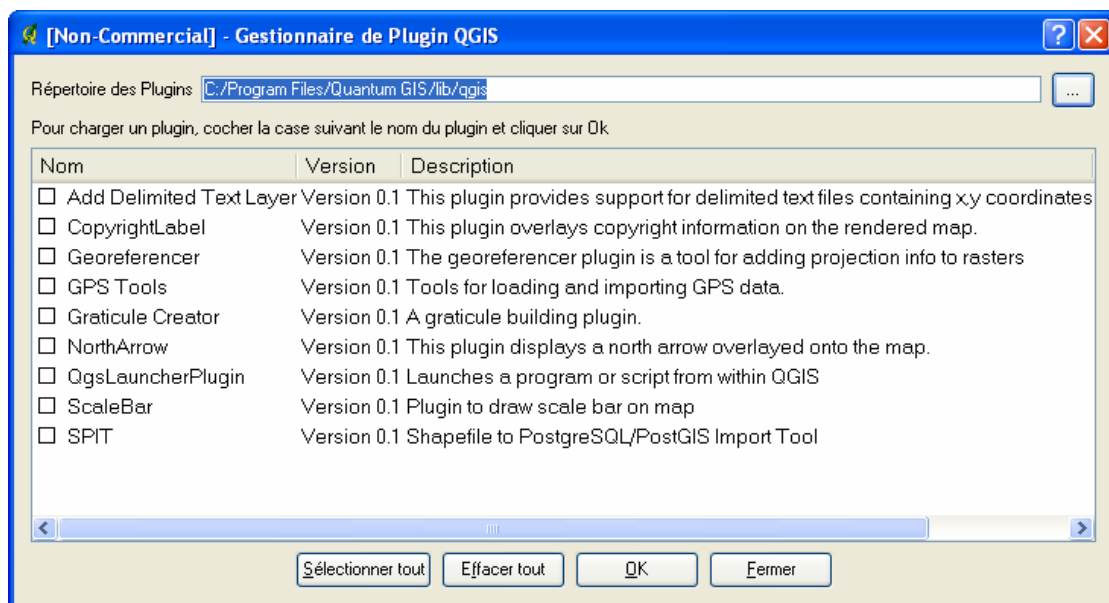
L'analyse thématique apparaît donc désormais sur la couche :



3) Les Plugins

QGIS possède en natif une série de plugins permettant diverses actions.

Nous ouvrons le gestionnaire des plugins, dans le menu « Plugin »



Le gestionnaire des plugins affiche la liste et la description de tous les plugins. Ce sont des plugins assez sommaires pour la plupart :

- Ajout des attributs de la carte : les plugins North Arrow et Scalebar, permettent d'afficher l'orientation et l'échelle de la carte.
- GPS Tools permet d'intégrer des point GPS
- Georeferencer est un outil de gestion des projections
- SPIT est un outil d'intégration de fichiers « shape » dans PostGIS
- QGS permet de lancer un programme ou un script depuis QGIS.
- Copright rajoute simplement un copright en bas à droite de la carte

Pour lancer un plugin, nous cochons la case correspondante et validons par « Ok ». Son icône s'ajoute automatiquement dans la barre d'outil.

QGIS est donc un logiciel doté d'une interface claire et intuitive qui permet d'attaquer facilement une base de donnée externe PostGIS. Il constitue donc un des SIG « open source » les plus accessibles pour la gestion à distance des bases de données. En revanche on peut déplorer la faiblesse des fonctionnalités développées.

II. Guide d'utilisation de OpenJUMP

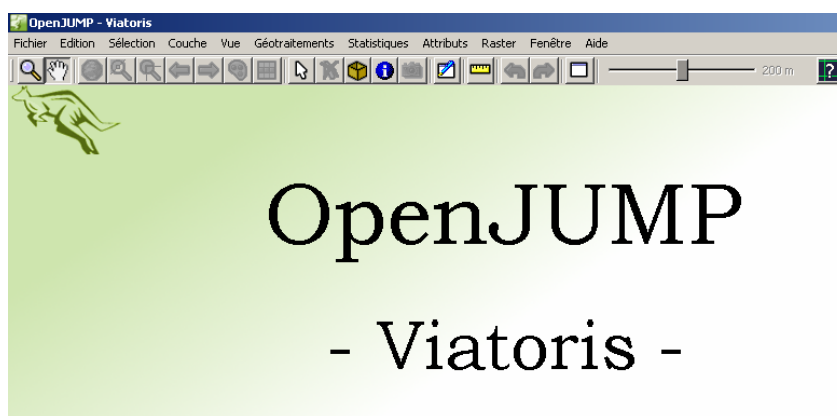


OpenJUMP
Version 2.0 du 29/01/2006

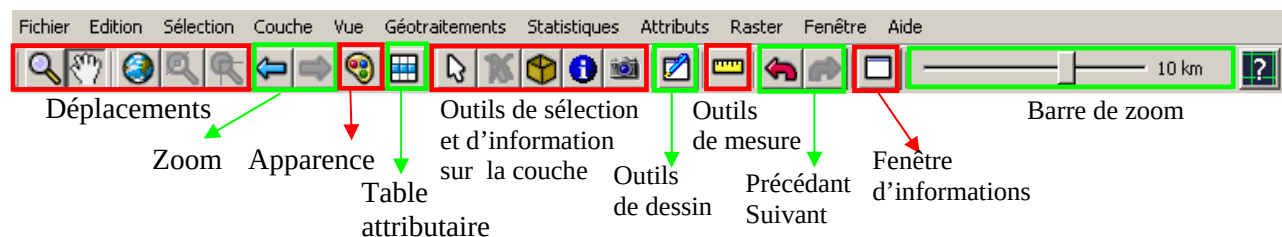


A. L'interface

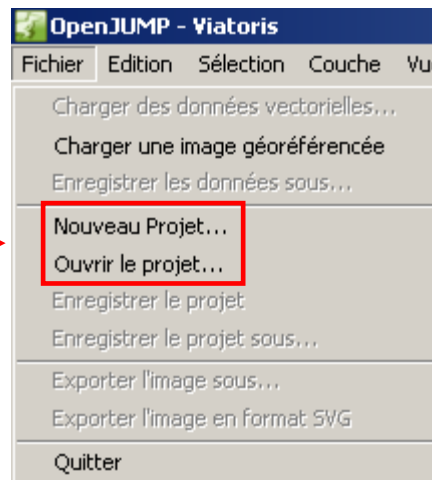
L'interface d'OpenJUMP est assez conviviale. Une fenêtre générale s'affiche à l'ouverture du logiciel.



Une barre d'outils « par défaut » rassemble toutes les fonctionnalités du logiciel, au travers de boutons :



Pour travailler sur un projet existant ou à créer, faire Fichier
Nouveau projet
ou
Ouvrir le projet

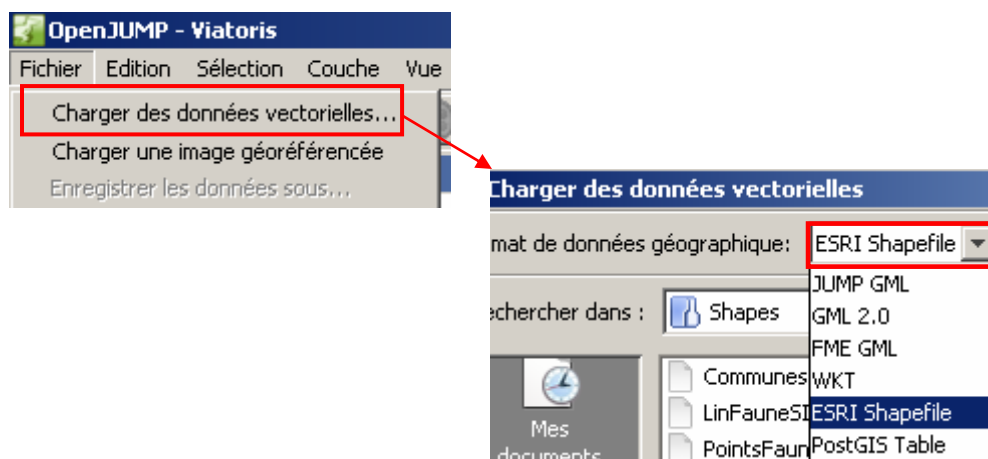


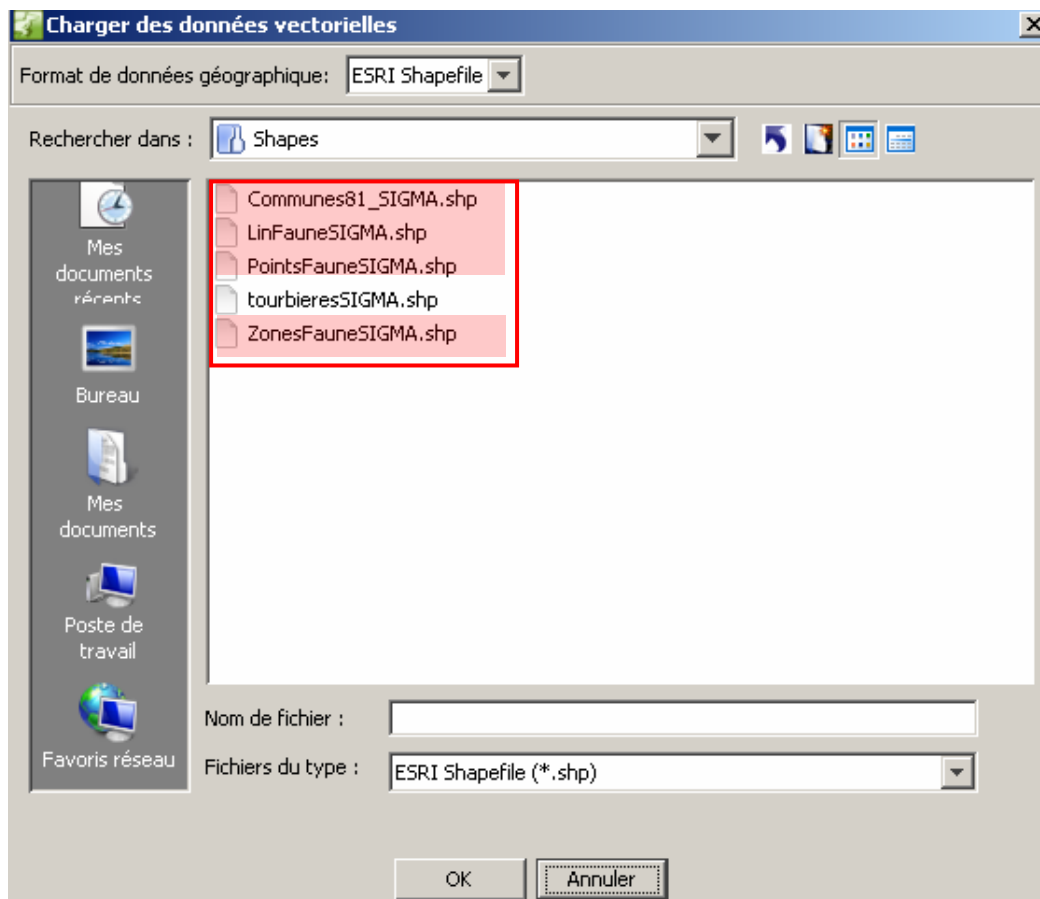
B. Import / export de Fichiers « Shape » (.shp)

Nous avons choisi de commencer les tests par l'import de données « shape » qui nous sont fournis par le CREN.

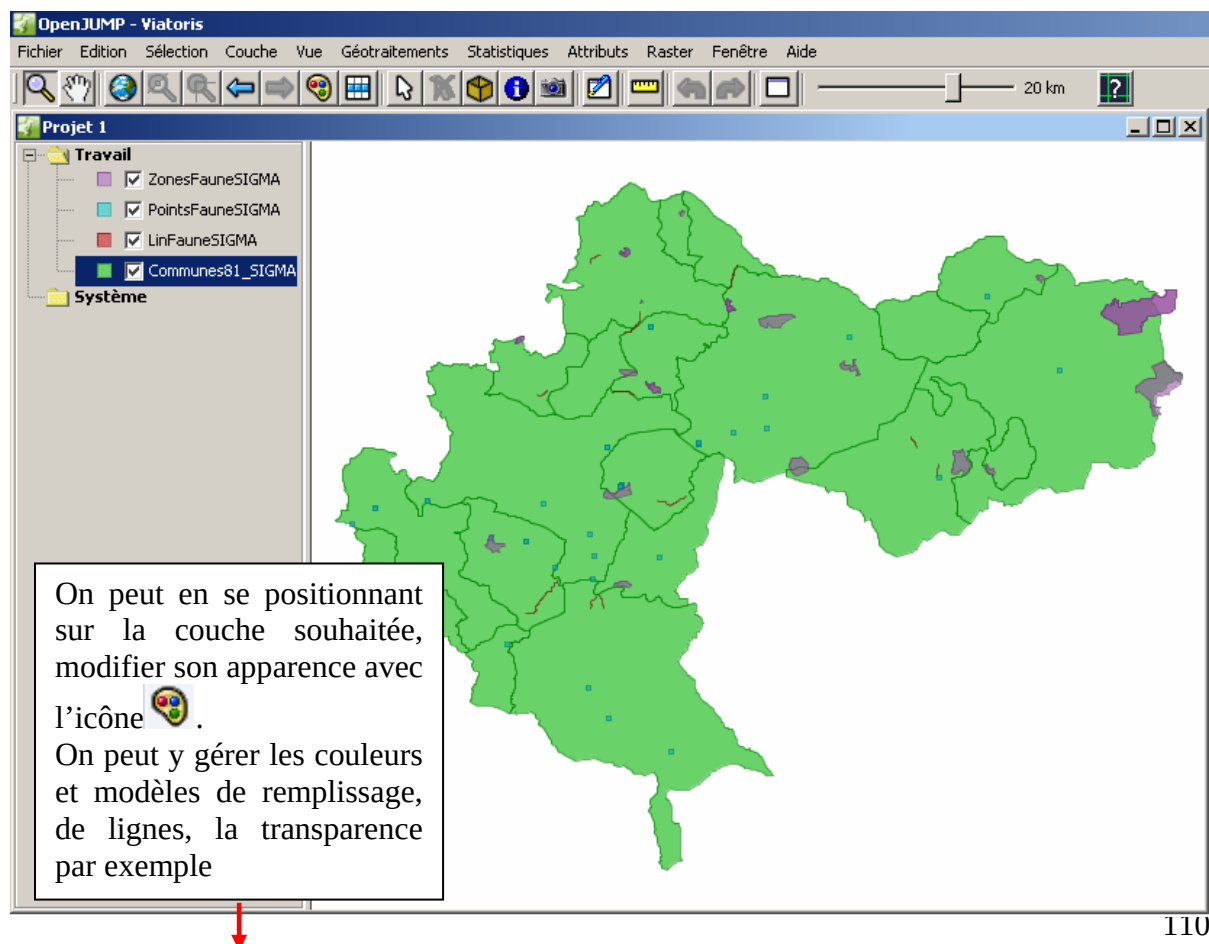
Il s'agit des fichiers communes81_SIGMA, LinFauneSIGMA, PointsFauneSIGMA, et ZonesFauneSIGMA

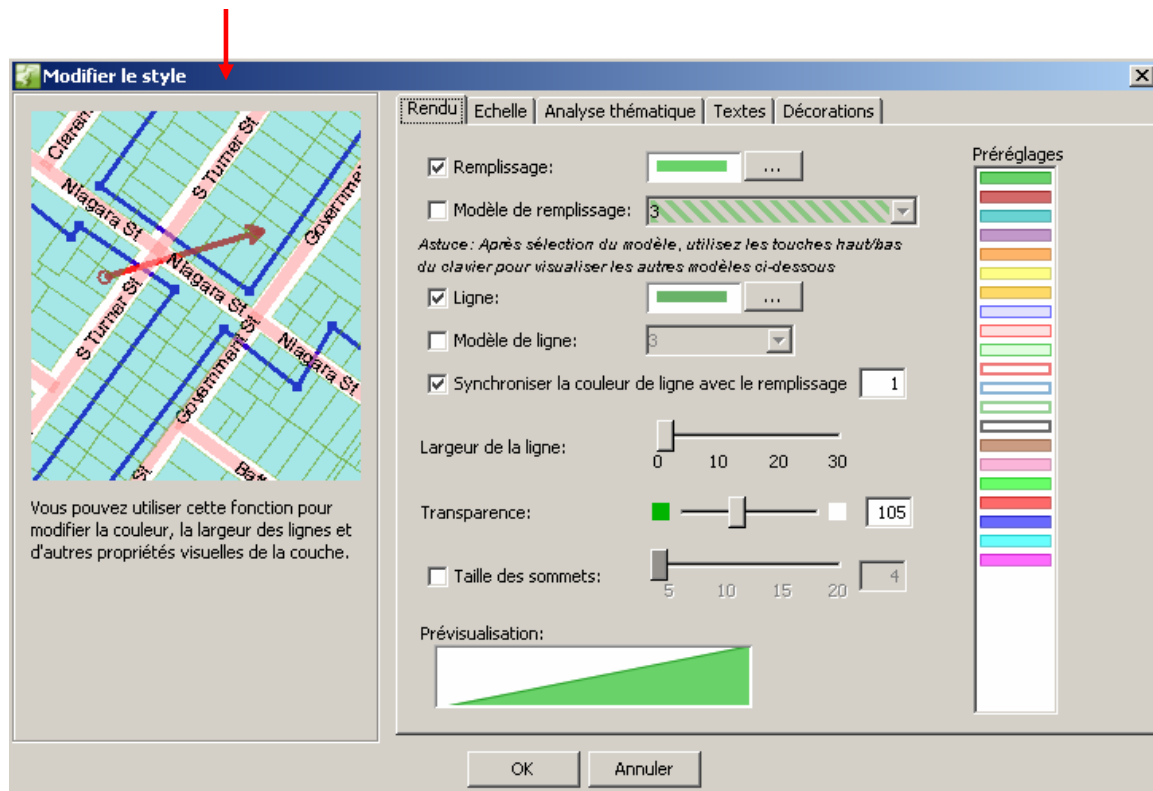
* l'import se fait par l'onglet fichier, charger des données vectorielles, choisir dans l'onglet format de données « ESRI Shapefile », puis les fichiers souhaités





Les couches sont maintenant chargées dans le projet :





La compatibilité avec la gamme ESRI est totale puisqu'il est aussi possible d'exporter en format shape (« Fichier / Enregistrer les données sous... » : il faut choisir le format « ESRI Shapefile » dans le haut de la fenêtre et donner l'extension .shp).

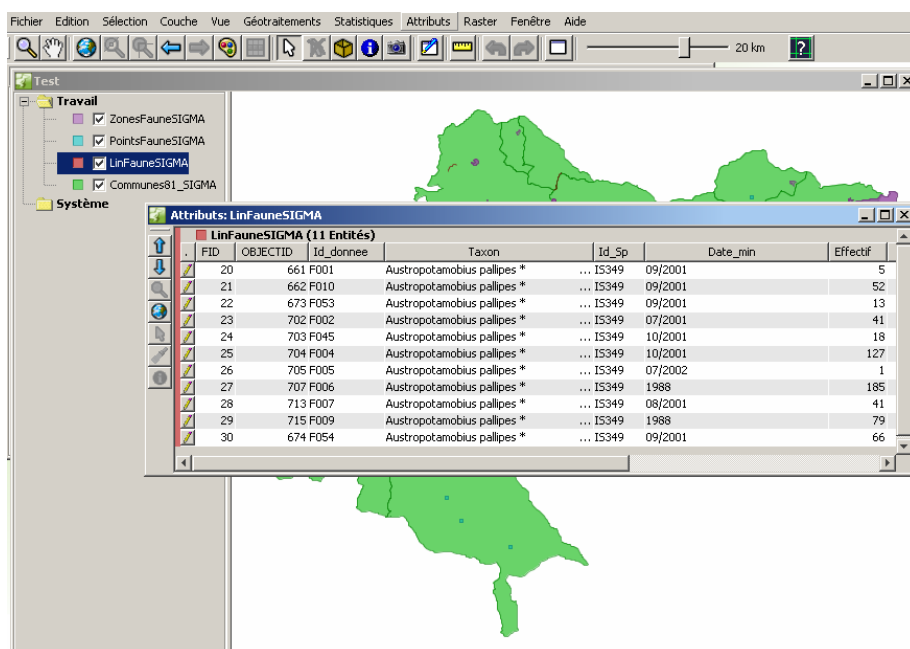


C. Consultation des données attributaires - Structure des tables

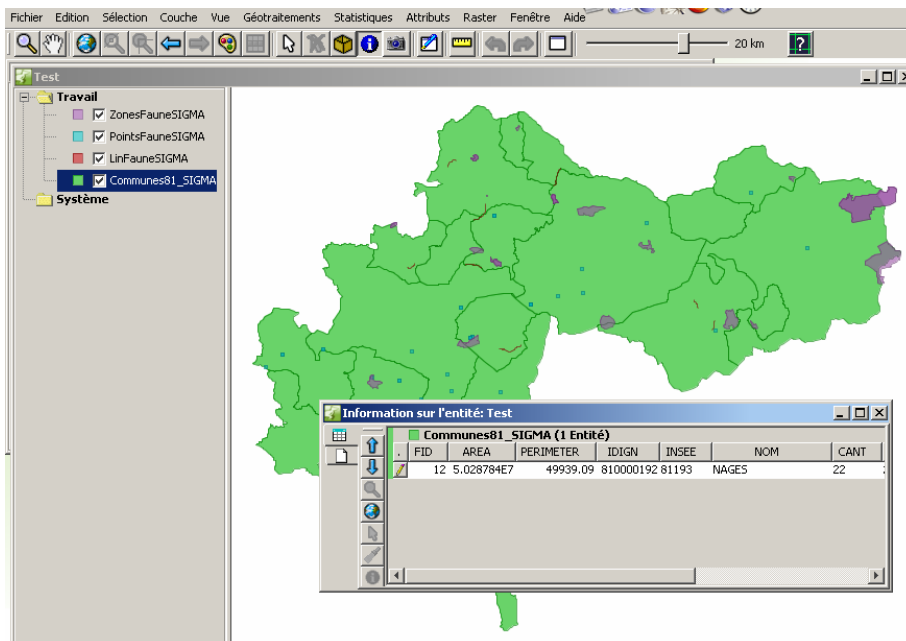
1) Visualisation des données

Une fois les couches chargées, il est possible de **consulter** leurs données attributaires :

* soit, en étant positionné sur la couche, par l'icône  qui donne les informations de **toute la couche** (tous les enregistrements)

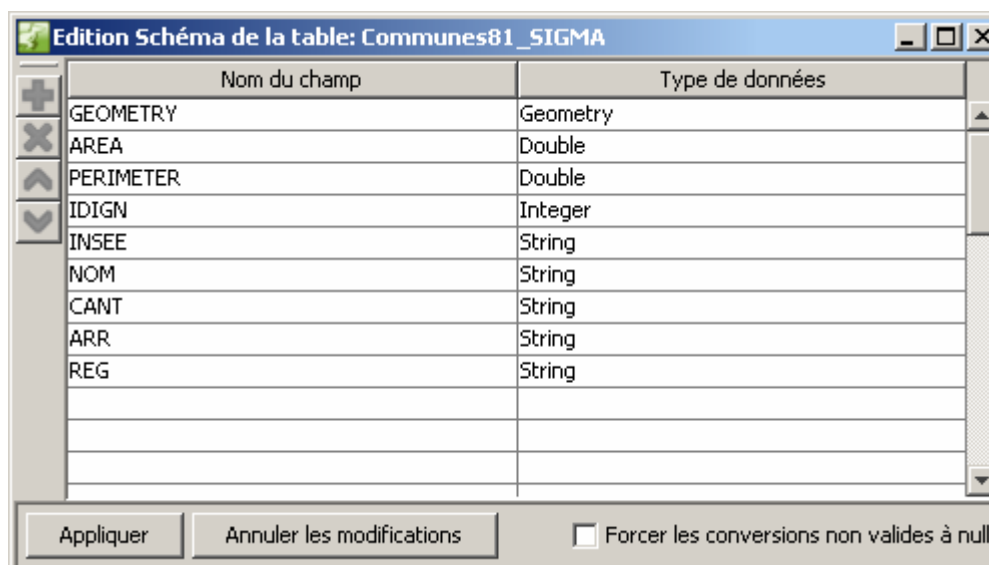


* soit, par l'icône  qui, si on le positionne sur une entité, permet d'afficher les informations liées strictement à cette entité sur cette couche.



2) Structure des tables

Nous pouvons accéder à la structure d'une table, en se positionnant sur la couche, clic droit et clic sur l'icône



S'affichent alors les différents champs de la table concernée, leur nom et leur format.



D. Edition des entités géographiques

Cette fonctionnalité est réalisée à partir de l'icône



Cette icône ouvre une boîte de dialogue, rassemblant tous les outils d'édition proposés



Sélection et déplacement d'entités

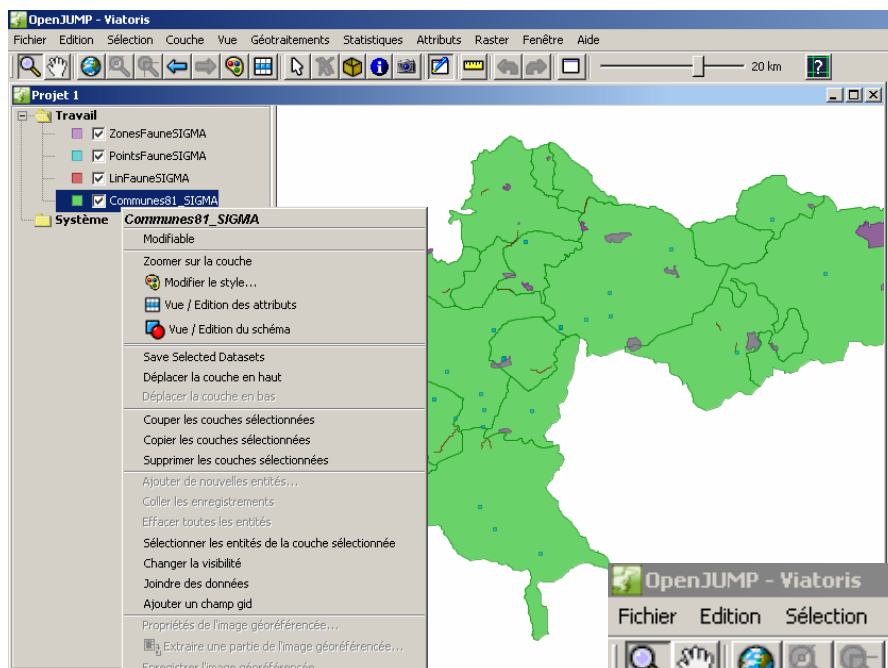
Outils de dessin (polygone, ligne, point)

Insertion, Suppression, déplacement d'un sommet

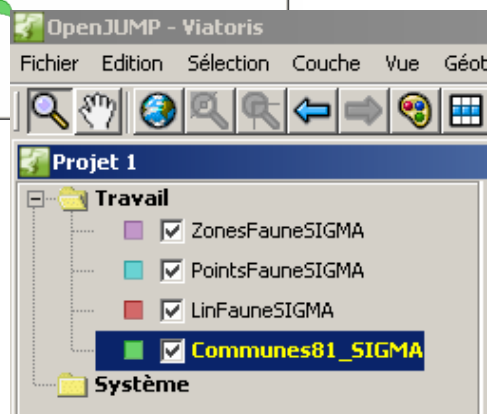
Edition d'entités contraintes



Avant toute modification, il est indispensable de rendre « **modifiable** » la couche concernée. Pour cela un clic droit sur la couche permet d'accéder à cette fonction : cliquer sur *modifiable*



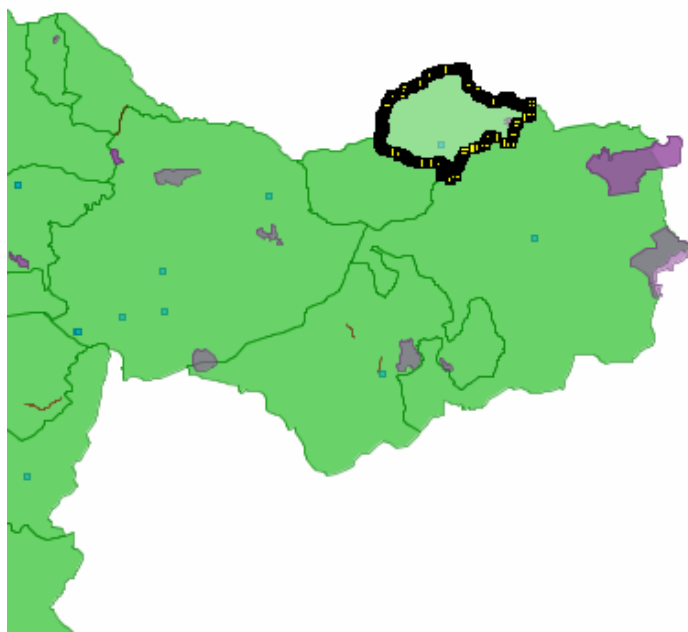
La couche est alors surlignée en jaune →



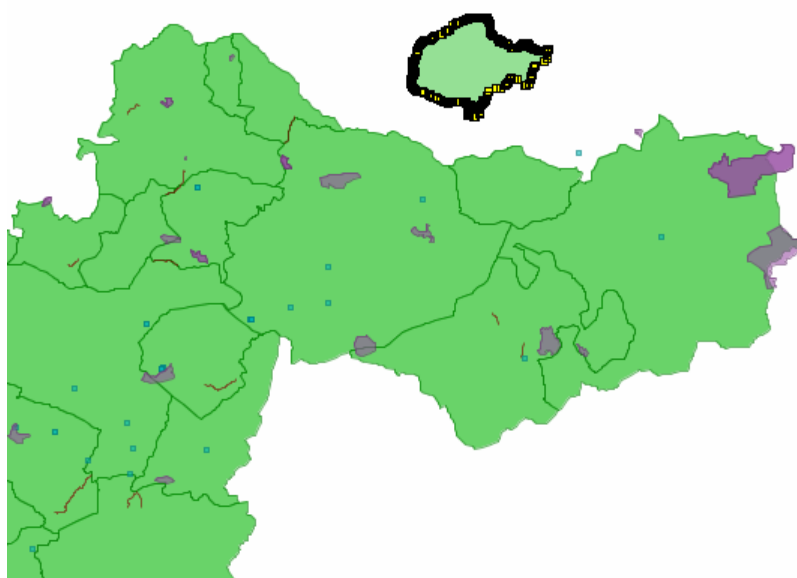
1) Sélection et déplacement d'entité



Sélection : Après avoir cliqué sur l'onglet  nous allons sur l'entité choisie (ici un polygone), un clic sur cette entité permet de la faire ressortir, en affichant les différents sommets.



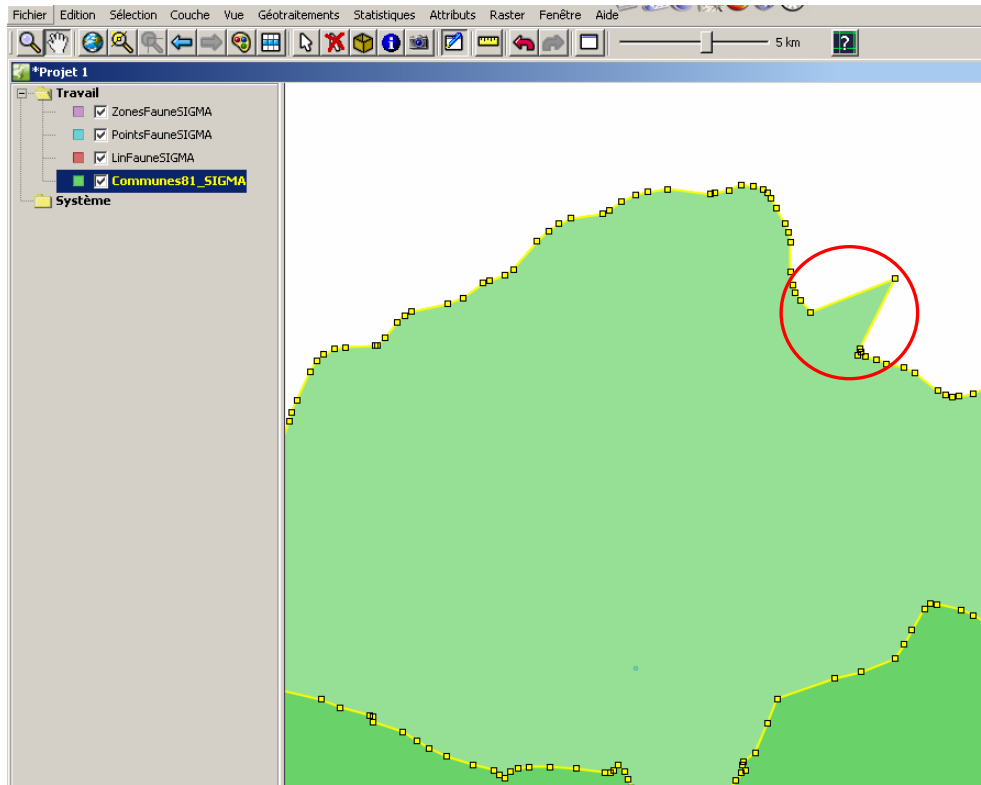
Déplacement : en cliquant sur l'outil  nous pouvons déplacer l'entité sélectionnée



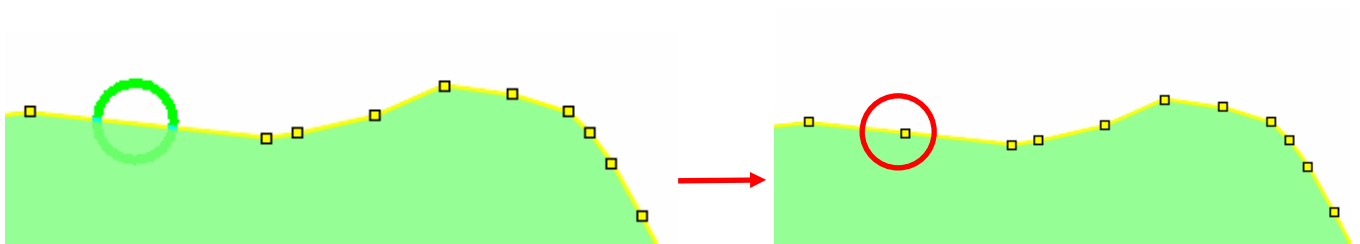
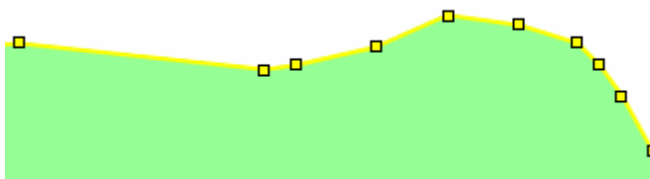
2) Déplacement, ajout et suppression de sommet



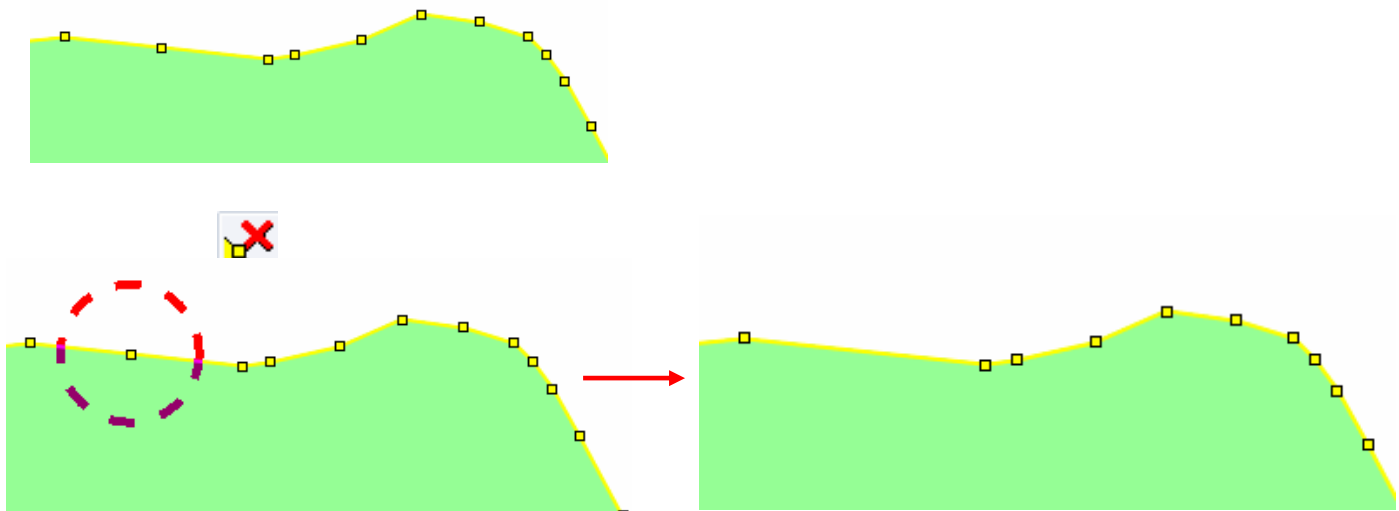
Déplacement : une fois l'entité sélectionnée, en cliquant sur  nous pouvons déplacer un sommet de l'entité



Ajout : Nous pouvons ajouter un sommet avec l'outil 



Suppression : de la même manière, nous pouvons supprimer un objet avec l'outil 



3) Saisie de lignes, points et polygones



Saisie : Nous pouvons saisir les 3 types d'entités graphiques (point, ligne, polygone) grâce à l'outil spécifique



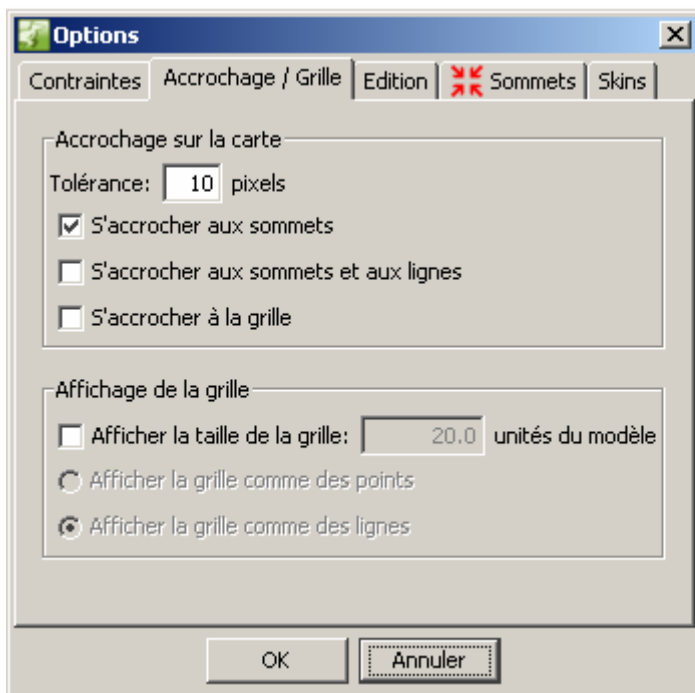
Pour un polygone



Pour une ligne



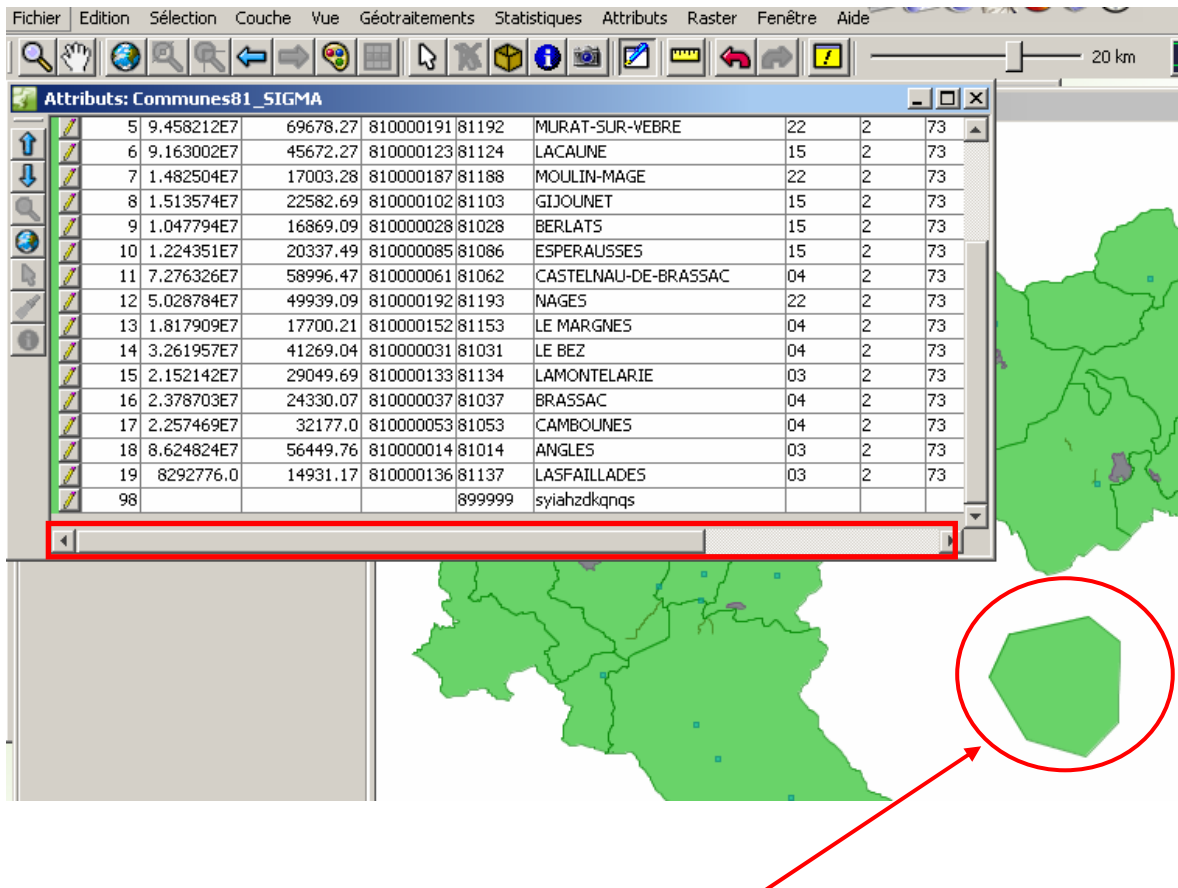
Pour un point



Pour chacune de ces saisies il est possible par l'onglet « options » de définir des propriétés de saisie (contraintes de saisie, accrochage, etc...)

4) Lien données géographiques - données attributaires

Quand nous saisissons une nouvelle entité, un enregistrement est automatiquement ajouté dans la table attributaire



Par exemple ici, nous avons saisi un nouveau polygone dans la couche « communes81 ». En ouvrant la table, nous pouvons saisir les informations sur la ligne correspondant à cet enregistrement.

Néanmoins, persistent des problèmes, notamment sur le calcul de surface, aussi bien sur des nouveaux polygones que sur des entités modifiées. En effet, OpenJUMP ne permettrait pas un affichage dynamique de cette surface. La mise à jour semblerait donc compromise.



E. Les requêtes et analyses thématiques

OpenJUMP offre la possibilité de faire des requêtes sur les données disponibles. Grâce à l'assistant de requêtes (*Sélection, Assistant de requête*), nous accédons à une fenêtre proposant les différentes possibilités

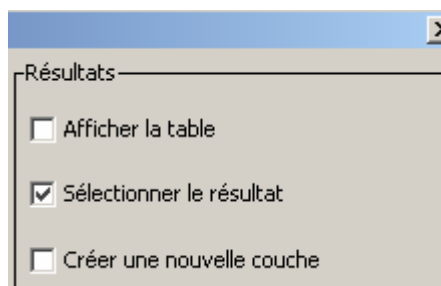
The screenshot shows the OpenJUMP - Viatoris application window. The 'Sélection' menu is open, showing options like 'Sélectionner les entités dans le cadre', 'Annuler la sélection', 'Sélectionner les entités dans un polygone', 'Sélectionner les entités dans un cercle', 'Sélectionner les entités de la couche', 'Select By Type...', and 'Assistant de requêtes'. The 'Assistant de requêtes' option is highlighted. Below the menu is the 'Constructeur de requête' dialog box. It has a 'Gestion de requêtes' section with 'Ouvrir' and 'Enregistrer' buttons. A 'Filtrer selon le type d'attribut' section has checkboxes for 'Texte', 'Numérique', 'Géométrique', and 'Sensible à la casse'. The 'Résultats' section has checkboxes for 'Afficher la table', 'Sélectionner le résultat', and 'Créer une nouvelle couche'. The main area of the dialog box has five columns: 'Couche', 'Attribut', 'Fonction', 'Opérateur', and 'Valeur'. Each column has a dropdown menu. Below the dialog box, five detailed dropdown lists are shown, corresponding to the columns: 'Couche', 'Attribut', 'Fonction', 'Opérateur', and 'Valeur'. Each list shows the current selection and a scrollable list of options.

Couche	Attribut	Fonction	Opérateur	Valeur
Toutes les couches	(GEOMETRY)		intersection	Toutes les couches
Toutes les couches	(GEOMETRY)	longueur	intersection	Toutes les couches
Sélection	AREA (DOUBLE)	surface	contient	Sélection
Couches sélectionnées	ARR (STRING)	nombre de points	est inclut dans	Couches sélectionnées
ZonesFauneSIGMA	CANT (STRING)	nombre de parties	est dedans (1000.0)	ZonesFauneSIGMA
PointsFauneSIGMA	Date_min (STRING)	buffer (1000.0)	touche	PointsFauneSIGMA
LinFauneSIGMA	Effectif (INTEGER)	centre	croise	LinFauneSIGMA
Communes81_SIGMA	IDIGN (INTEGER)	est vide	recouvre	Communes81_SIGMA
	INSEE (STRING)		est disjoint	

Après avoir structurer la requête, nous pouvons paramétrer le résultat voulu dans la fenêtre

Nous pouvons ainsi choisir

- * l'affichage de la table résultante
- * sélectionner le résultat dans la table correspondante
- * Créer une nouvelle couche avec les résultats obtenus



1) Exemple d'une requête simple

Nous voulons faire une requête de sélection de toutes les entités de la couche « Commune81 » dont le code « IINSEE » est le 810...

Le tableau de requête se présente comme suit et l'on demande l'affichage de la table correspondante et la sélection des entités résultantes

Constructeur de requête

Gestion de requêtes: Ouvrir, Enregistrer

Filtrer selon le type d'attribut: ☒ Texte, ☐ Sensible à la casse, ☒ Numérique, ☒ Géométrique

Résultats: ☒ Afficher la table, ☒ Sélectionner le résultat, ☐ Créer une nouvelle couche

Couche: Couches sélectionnées, Attribut: INSEE (STRING), Fonction: , Opérateur: commence par, Valeur: 810

Valider, Rafraîchir, Annuler, Stopper

Le résultat donne 9 entités qui s'affichent dans une table spécifique et ressortent sur la carte

Constructeur de requête

Gestion de requêtes: Ouvrir, Enregistrer

Filtrer selon le type d'attribut: ☒ Texte, ☐ Sensible à la casse, ☒ Numérique, ☒ Géométrique

Résultats: ☒ Afficher la table, ☒ Sélectionner le résultat, ☐ Créer une nouvelle couche

Couche: Couches sélectionnées, Attribut: INSEE (STRING), Fonction: , Opérateur: commence par, Valeur: 810

Sélectionner depuis "Couches sélectionnées" entités où INSEE commence par "810" : 9 entités trouvées

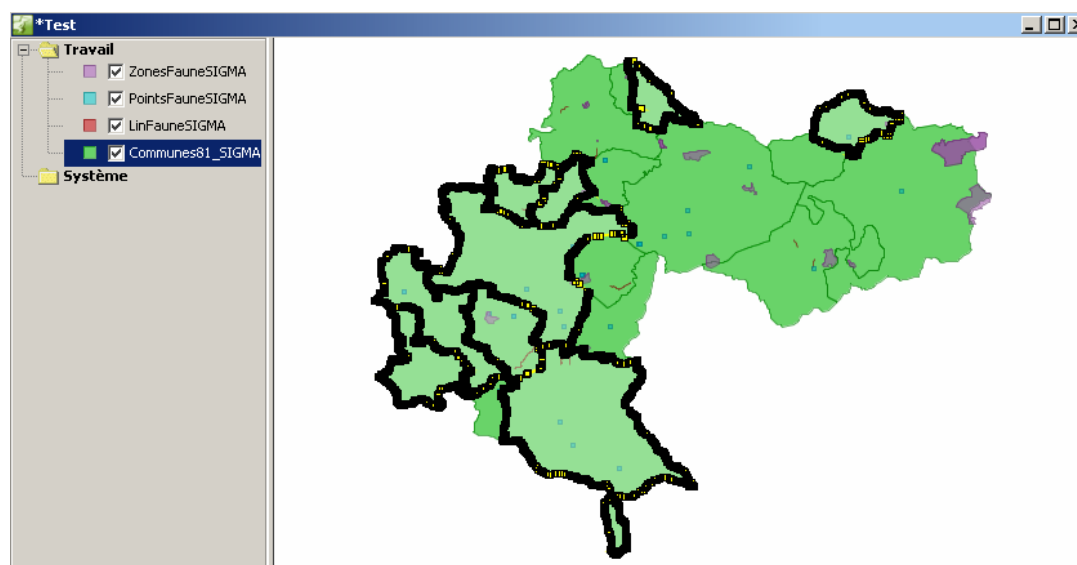
Valider, Rafraîchir, Annuler, Stopper

Information sur l'entité: Test

Communes81 SIGMA (9 Entités)

FID	AREA	PERIMETER	IDIGN	INSEE	NOM	CANT	ARR	REG
18	8.624824E7	56449.76	810000014	81014	ANGLES	03	2	73
4	1.51067E7	18684.2	810000023	81023	BARRE	22	2	73
9	1.047794E7	16869.09	810000028	81028	BERLATS	15	2	73
14	3.261957E7	41269.04	810000031	81031	LE BEZ	04	2	73
16	2.378703E7	24330.07	810000037	81037	BRASSAC	04	2	73
17	2.257469E7	32177.0	810000053	81053	CAMBOUNES	04	2	73
11	7.276326E7	58996.47	810000061	81062	CASTELNAU-DE-BRASSAC	04	2	73
3	1.026191E7	18601.68	810000084	81085	ESCROUX	15	2	73
10	1.224351E7	20337.49	810000085	81086	ESPERAUSSES	15	2	73

Nous observons ici les données géographiques sélectionnées avec la requête :

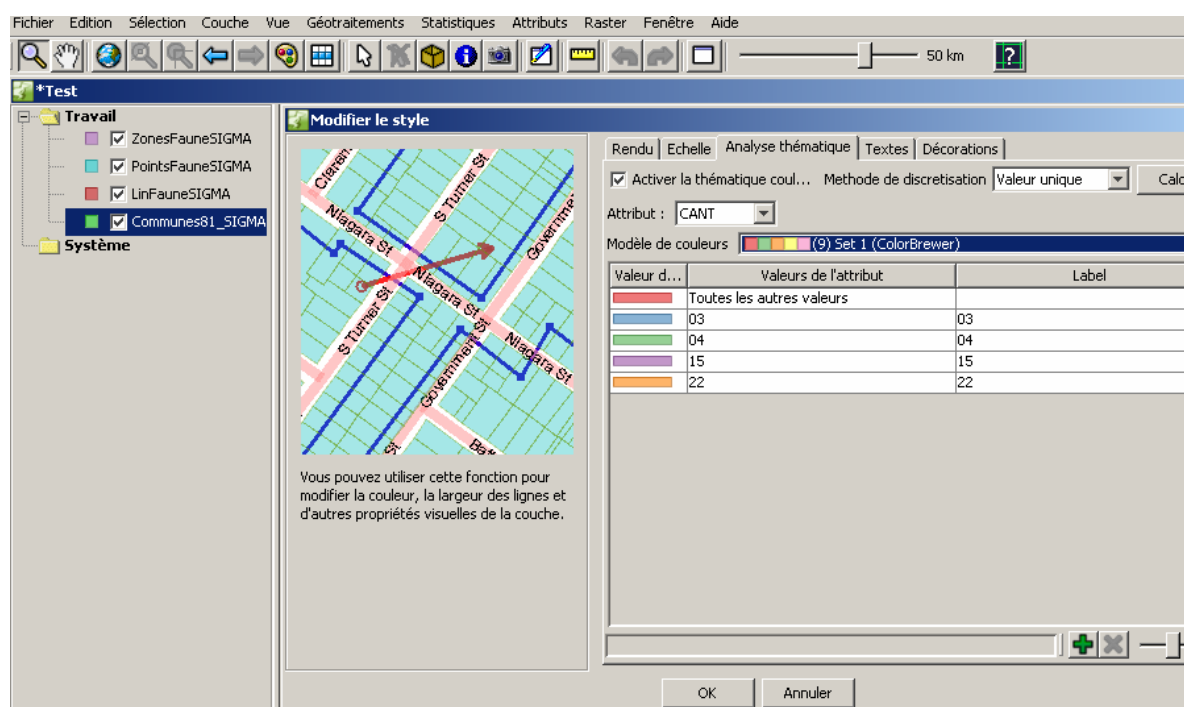


2) Exemple d'une analyse thématique

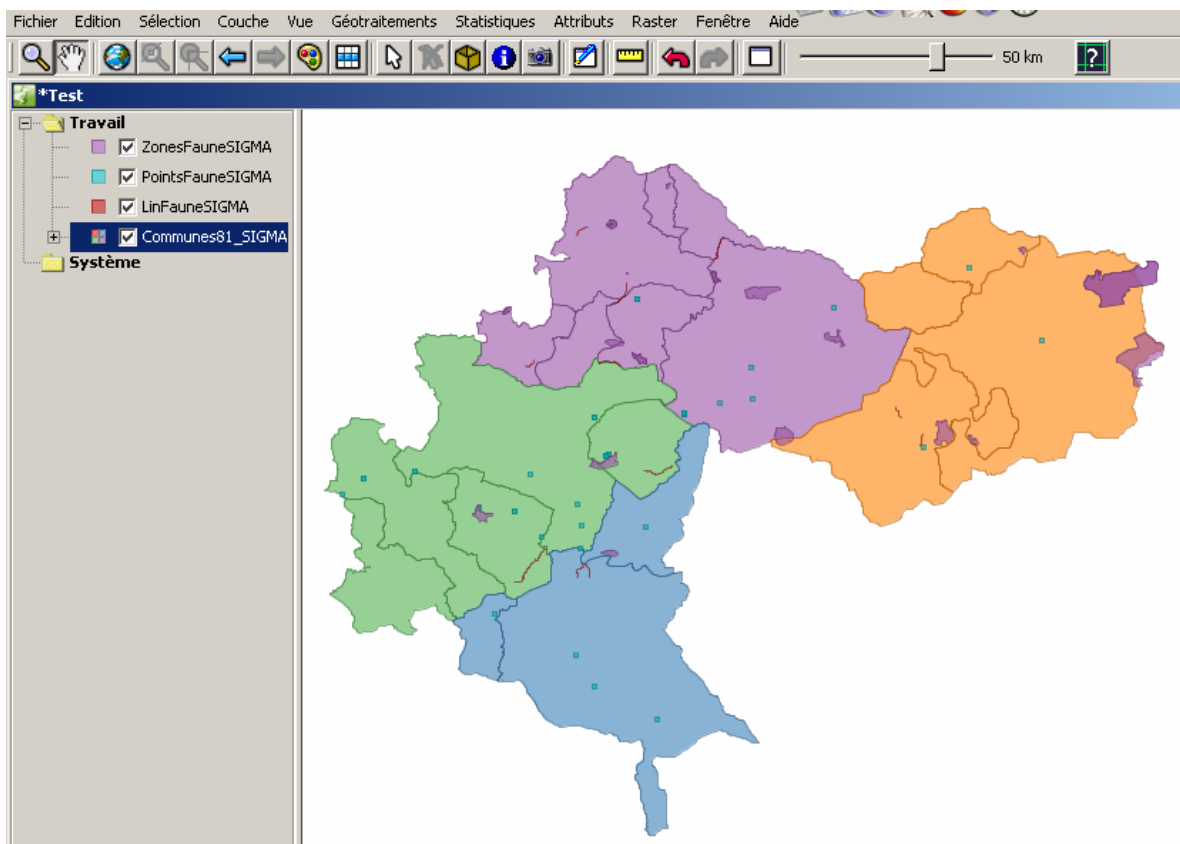
Nous voulons par exemple faire une analyse thématique sur le code canton de la couche « Communes81 ».

Nous accédons à la fenêtre concernée par le bouton 

Dans l'onglet « analyse thématique », nous activons la thématique couleurs sur le code canton (CANT)

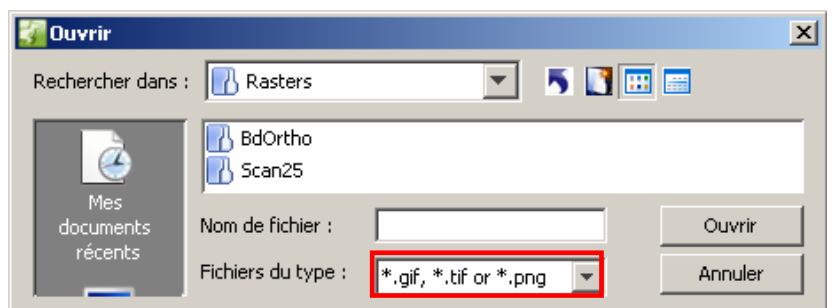
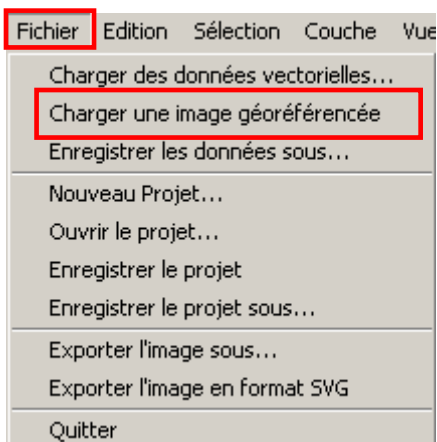


Après avoir lancé l'analyse, nous obtenons la carte thématique suivante :



F. L'import de fichiers Raster

De la même manière que nous avons importé des fichiers vecteurs, nous pouvons importer des fichiers raster (image). Toutefois, seuls des fichiers de format *.tif*, *.gif*, *.png* peuvent être importés. Ici, nous allons importer des dalles de Scan25 ®

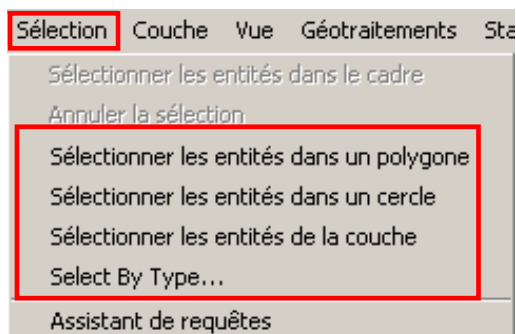




G. Fonctionnalités supplémentaires d'OpenJUMP

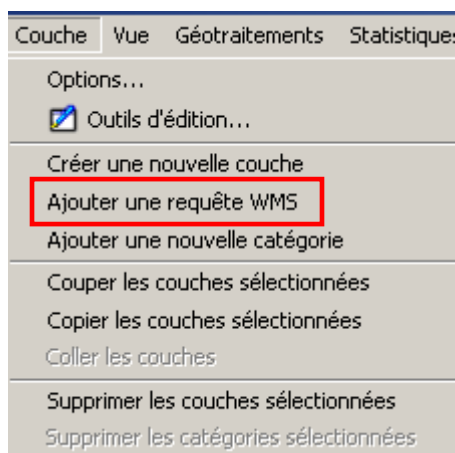
Hormis les propriétés décrites ci-dessus, OpenJUMP propose de nombreuses fonctionnalités. Quelques sont évoquées ici

1) Options de sélection



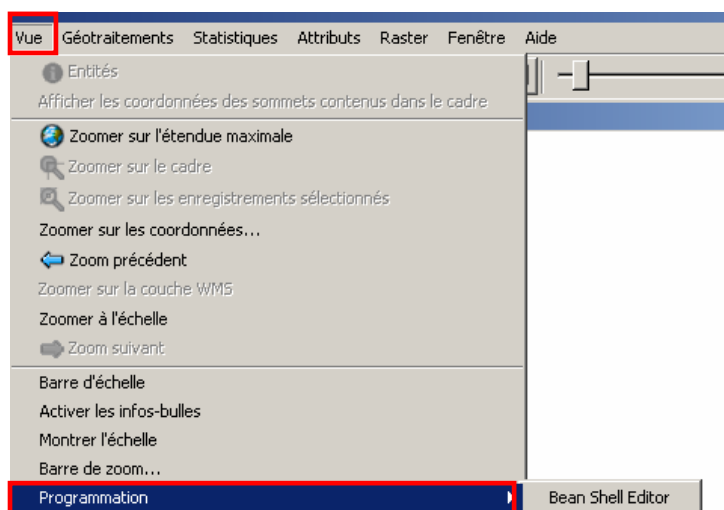
On peut paramétrer les sélections de manière à ne choisir qu'une partie des données recherchées

2) Options des couches



Parmi les diverses fonctionnalités offertes par OpenJUMP, il est intéressant de pouvoir charger des données géographiques stockées sur un serveur distant grâce à la technologie WMS (**WebMapService**)

3) Programmation



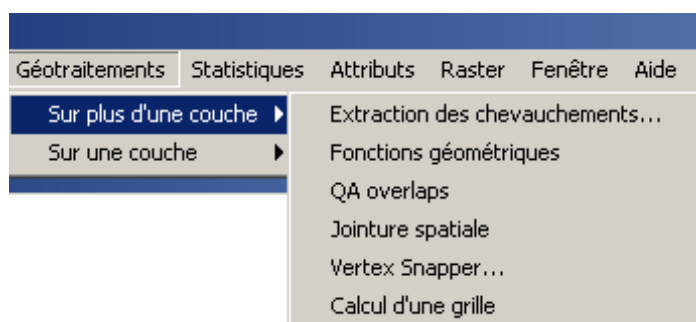
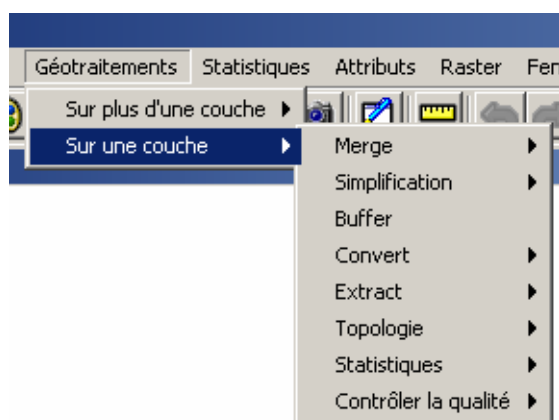
Il est possible d'étendre les fonctionnalités d'OpenJUMP avec des extensions (ou « plugins »)
Notamment l'une d'entre elles, « Bean Shell Editor » (version 0.1.1)

BeanShell Editor est une interface qui permet d'écrire des scripts BeanShell pour piloter OpenJUMP. Vous pouvez par exemple :

- réaliser des opérations géométriques entre des entités géographiques,
- changer le style d'une couche,
- remplacer des valeurs d'un champ d'une couche...

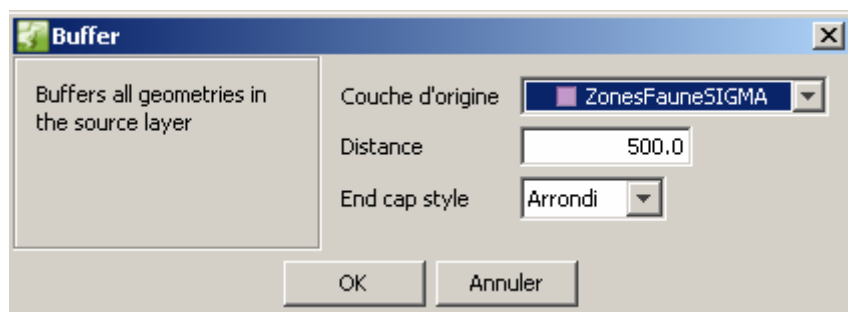
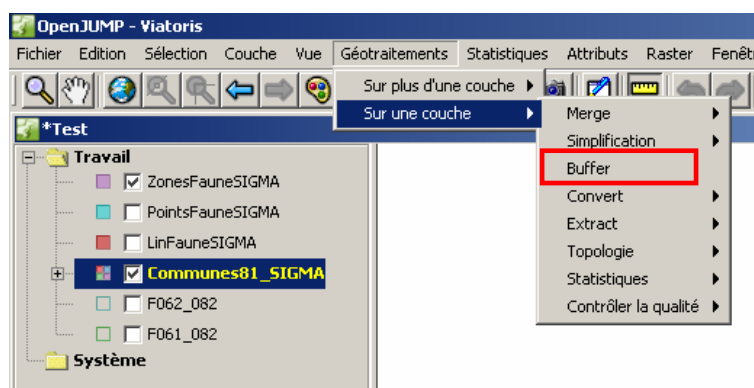
4) Géotraitements

Il est possible d'effectuer des géotraitements sur une ou plusieurs couches



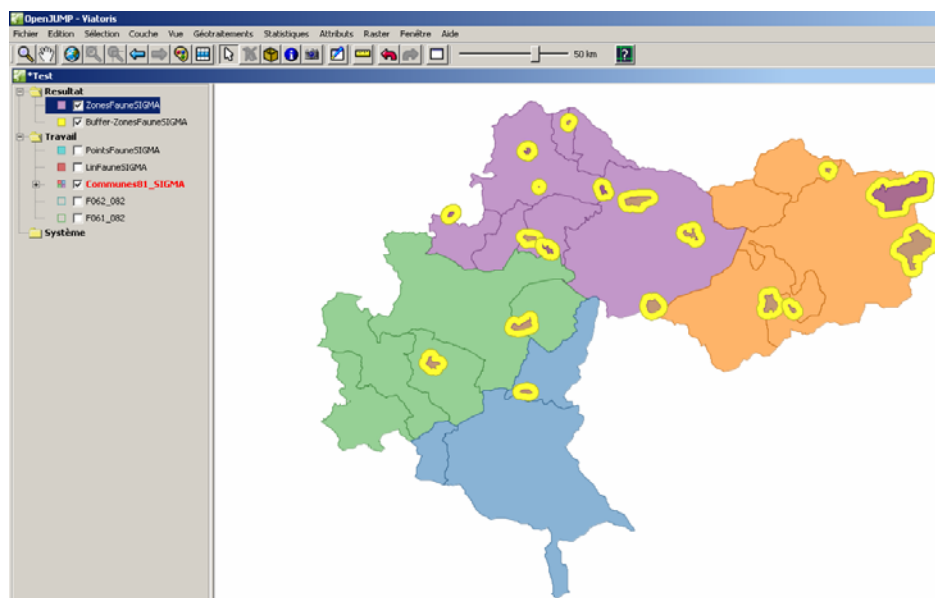
Exemple : Nous voulons réaliser un « Buffer » (tampon) de 500 mètres autour des entités ZonesFaune

L'option « Géotraitements » sur la couche concernée, puis sur **Buffer** ouvre une autre fenêtre dans laquelle nous choisissons la couche, la distance (ici 500 m) et le style (ici arrondi) du Buffer



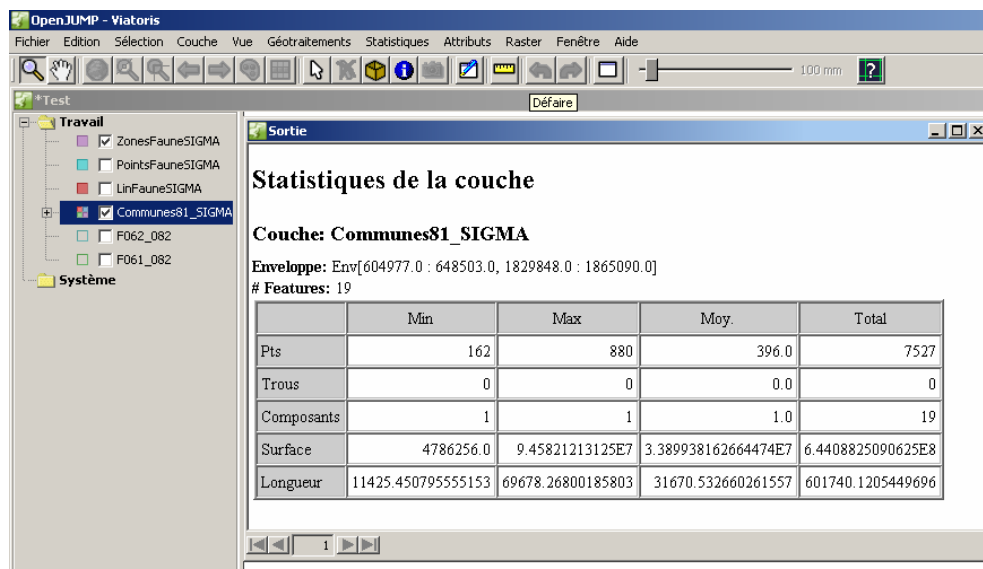
La carte résultante est la suivante :

Un répertoire « Résultats » comportant la couche « Buffer » a été rajouté et les zones tampon apparaissent sur la carte.

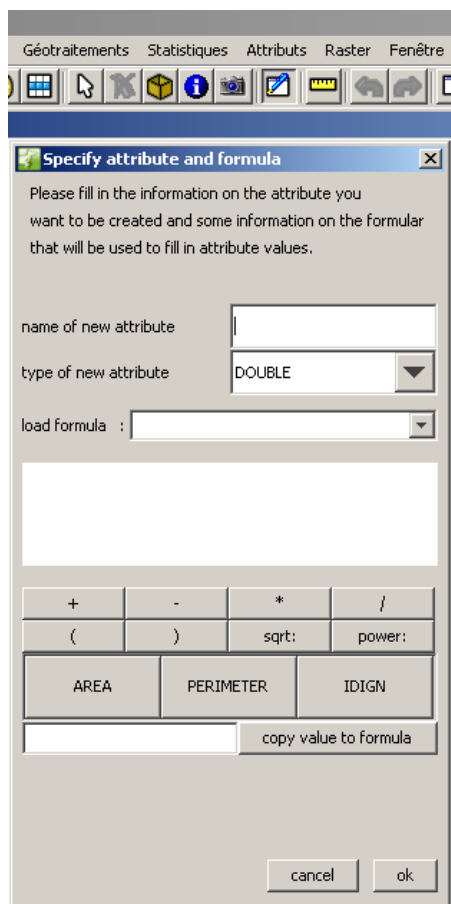


5) Statistiques

Nous pouvons afficher les statistiques propres à chaque couche.



6) Editeur de formules



7) Options des couches

Pour chacune de couches du projet, il est possible d'accéder à diverses fonctionnalités

